A Parallel Manipulator for Prostate Cancer Biopsies:

Kinematic Optimization, Design and Control Planning

Catarina Salomé Dias Carvalheiras

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisor: Prof. Jorge Manuel Mateus Martins

Examination Committee

Chairperson: Prof. Paulo Jorge Coelho Ramalho Oliveira
Supervisor: Prof. Jorge Manuel Mateus Martins
Member of the Committee: Prof. João Carlos Prata dos Reis

June 2017

A robot may not injure a human being or, through inaction, allow a human being to come to harm.

Isaac Asimov

Acknowledgments

I would like to seize this moment to share my sincere gratitude to everyone who helped and encouraged me through the development of this work.

First and foremost, I would like to thank my university colleagues and dearest friends, Pedro Pontes, Vasco Rodrigues, João Oliveira, Ricardo Pedro and Henrique Carvalho, without whom I would've not made it this far. Thank you for your friendship and support since the beginning of this journey.

I would like to acknowledge my advisor, Prof. Jorge Martins, who supervised and guided me through this study, providing me knowledge, aid and feedback.

I also must thank Senhor Raposeiro and Engenheiro Camilo, from the ACCAII laboratory, for all the patience and help that made the concretization of this thesis possible. I would like to thank Dona Mónica for all the toasts and coffee after some rather restless nights.

Last but not least, I thank my parents, my sister and my whole family for their unconditional love, comprehension and support.

Resumo

Neste trabalho o desenvolvimento de um protótipo de um robot paralelo é apresentado. Este protótipo foi desenvolvido para ser implementado com uma sonda de ultra sons transrectal para biópsias à próstata, sendo co-manipulado por um urologista. De toda a mortalidade devida a cancro, o cancro da próstata é a sexta causa mais comum, representando 9% de todos os óbitos masculinos. Os métodos mais comuns para diagnóstico são análises aos níveis do antígeno prostático específico, e o exame rectal digital. Se estes testes revelarem indícios de cancro, é aconselhada uma biópsia à próstata guiada por ultra sons. Em muitos casos, o médico pode falhar o tecido cancerígeno o que resulta num teste falsamente negativo. O objectivo do mecanismo deste trabalho é guiar o médico até aos pontos óptimos de perfuração, através de controlo de trajectória. Este trabalho foca-se na optimização cinemática, baseada em dados recolhidos, de um controlador háptico comercial, o Novint Falcon. Este manipulador foi desmontado, optimizado e reconstruído. Os objectivos de optimização foram a maximização da manipulabilidade cinemática, e o máximo número de pontos de trajectória alcançados. Estes pontos pertenciam a dados recolhidos durante biópsias à próstata, previamente registadas. Os modelos geométrico, cinemático e dinâmico foram deduzidos. O desenho, manufactura e procedimentos de assemblagem são apresentados. De modo a obter um mecanismo controlável, também é referida a implementação de motores DC, encoders ópticos, filtros e processamento de sinal, e comunicação série entre uma placa Arduino e o programa Simulink. Um protótipo funcional foi construído com sucesso.

Palavras-chave: Biópsias à Próstata, Co-Manipulação, Robótica Paralela, Modelação Cinemática, Optimização Cinemática.

Abstract

The development of a prototype of a parallel robot is presented in this work. This prototype is to be implemented with a trans-rectal ultra sound probe for prostate biopsies, which is to be co-manipulated by an urologist. Prostate cancer is the sixth most common cause of cancer death overall, representing 9% of all male deaths. The most common methods for diagnosis are blood tests for scanning of the prostate-specific antigen levels, and digital rectal examination. If these tests reveal some signs of cancer suspicion, an ultra sound, manually guided biopsy to the prostate is usually advised. In many cases, the doctor can miss the cancerous tissue which will lead to a false negative test. The objective of the developed device is to guide the doctor to the optimal points of perforation through trajectory control. This work focuses on the data-based kinematic optimization of an existing commercial haptic controller, the Novint Falcon. This device was disassembled, optimized and rebuilt. The objectives of optimization were the maximization of kinematic manipulability and largest number of trajectory points reached from data sets acquired during previously registered prostate biopsies. The geometric, kinematic and dynamic models are deduced. The design after optimization, manufacture and assembly procedures are detailed. In order to achieve controllable mechanics, the implementation of DC motors, rotation encoders with disks, signal filtering and processing, and serial communication using an Arduino board and Simulink software are described. A functional prototype was successfully built.

Keywords: Prostate Biopsy, Co-Manipulation, Parallel Robotics, Kinematic Modelling, Kinematic Optimization.

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

ϕ	Joint referential angle to the base coordinate system
θ	Joint angles

Roman symbols

c_d	Viscous damping coefficient.
g	Gravity acceleration.
I	Mass moment of inertia.
m	Mass.

Subscripts

A_r	Reduced matrix
u, v, w	Joint cartesian components
x, y, z	Global cartesian components

Superscripts

$\ddot{x}$	Second derivative with respect to time.
$\dot{x}$	Derivative with respect to time.
x^*	Adjoint
x^T	Transpose
x^{-1}	Inverse

List of Acronyms

CAD	Computed Assisted Design
DC	Direct Current
DGM	Direct Geometric Model
DKM2	Direct Kinematic Model of Second Order
DKM	Direct Kinematic Model
DM	Dynamic Model
DOF	Degree of Freedom
I/O	Input/Output
IDM	Inverse Dynamic Model
IGM	Inverse Geometric Model
IKM2	Inverse Kinematic Model of Second Order
IKM	Inverse Kinematic Model
ISR	Interrupt Sub-routine
PWM	Pulse Width Modulation
SDK	Sofware Developper Kit
opAmp	Operational Amplifier

1 | Introduction

Because the objective of this work is to develop a co-manipulation robot for the prostate biopsy procedure, it is important for one to know why this is a urging problematic.

1.1 Motivation

The prostate is a gland of the male reproductive system. It surrounds the urethra and the two ejaculatory ducts and it is approximately (in normal conditions) the size and shape of a walnut as illustrated in Figure 1.1.

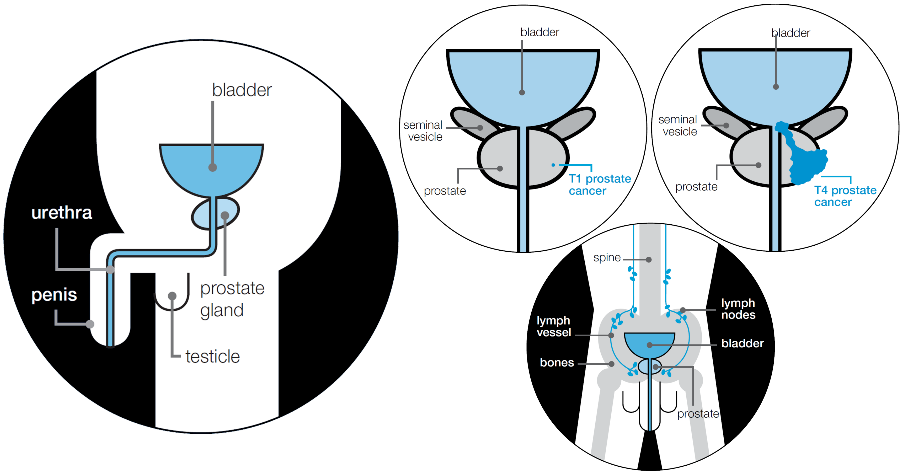
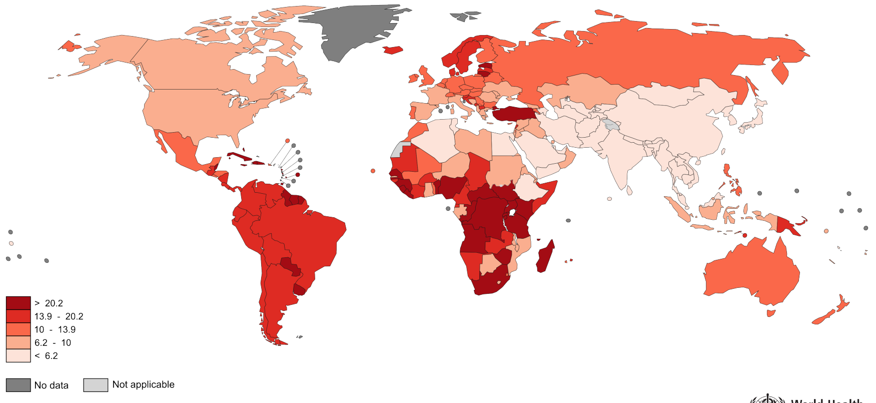


Figure 1.1: Prostate gland anatomy in the human body and various stages of prostate cancer. [1]

As an aging effect, the size and texture of the prostate gradually changes. It is common for men after 50 years of age to develop prostatic enlargement. Although this is not preventable, there are treatments advised to try and reverse the possible negative effects [2]. One of the most frequent degeneracy of this effect is prostate cancer. Being the 6th most common cause of cancer death overall, with around 307 000 mortality cases in 2012, it represents 6.6% of all male deaths [3]. Although it generally has a high survival rate when early diagnosed, the prostate cancer is the 5th more incident cancer worldwide, with a high incidence rate as shown in Figure 1.2.



The boundaries and names shown and the designations used on this map do not imply the expression of any opinion whatsoever on the part of the World Health Organization concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. Dotted and dashed lines on maps represent approximate border lines for which there may not yet be full agreement.

Data source: GLOBOCAN 2012
Map production: IARC
World Health Organization

World Health Organization
© WHO 2015. All rights reserved

Figure 1.2: Estimated prostate cancer mortality worldwide in 2012, per 100 000. [4]

1.2 Diagnosis

Prostate problems can cause urinary symptoms. These symptoms are generally not derived from cancer but from the enlargement of the prostate. This is why it is necessary to make a few tests in order to diagnose prostate cancer, which are: Urine test to rule out infection, Prostate specific antigen (PSA) blood test and Digital rectal examination with or without ultrasound probe.

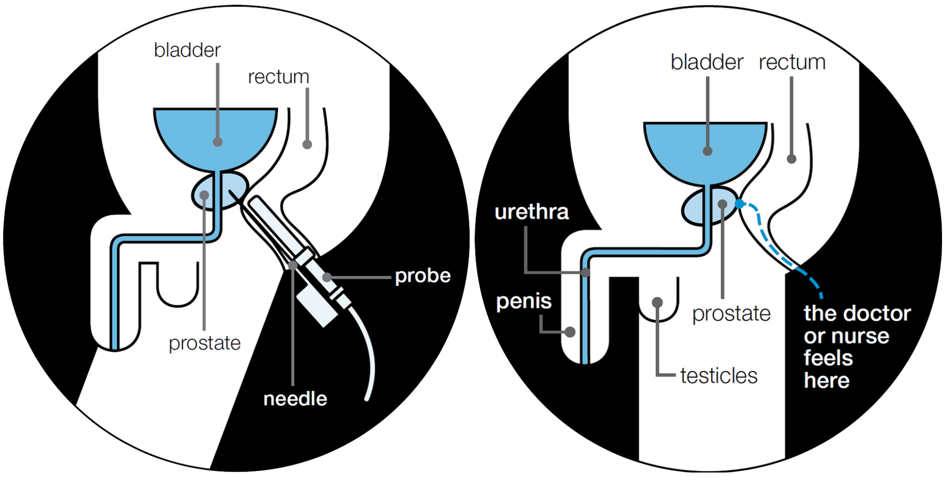


Figure 1.3: Prostate cancer diagnosis methods, the trans-rectal ultrasound guided biopsy on the left and the digital rectal examination on th right [1]

If these tests reveal some signs of cancer suspicion, usually the patient is advised to take a Transrectal Ultrasound (TRUS)-guided prostate biopsy to further diagnose as illustrated in Figure 1.3. However, recent studies show that the PSA test can be utterly misleading. This is because the PSA level can be affected if the man has engaged in physical activity or sex prior to the test. It is also possible that a positive PSA test will reduce the quality of life of a patient with a small cancer that might never grow large enough to cause problems during his lifetime. Also, many men can develop prostate cancer despite having low levels of PSA [5].

Because of the common misinterpretation of the PSA levels, many men suffer unnecessary TRUS-guided biopsies to the prostate and only after this are lead to an electronic scan process like a Magnetic Resonance Imaging (MRI) or a Computed Tomography Scan (CT). Usually CT scans are not as useful as a MRI to look at the prostate gland itself but are rather helpful to detect if the cancer has spread into nearby lymph nodes, as it was depicted in Figure 1.1.

With the increased availability of this technology, some doctors have started to prescribe MRI scans to diagnose early prostate cancer as a non intrusive method, before effectively resorting to the invasive biopsy of the prostate [6].

1.3 The Problem

Studies have shown that the current method of TRUS-guided biopsy has a big room for improvement. In his Master thesis dissertation, Tavares [7] reconstructed a 3D model of the prostates of 22 TRUS-guided biopsy patients in Hospital Curry Cabral who had previously been submitted to an MRI. With the MRI images, a 3D model was created prior to the TRUS-guided biopsy. During each one of the exams, an optical tracking system was used to track the position and orientation of the biopsy probe.

The TRUS-guided biopsy method used by the Urologist was to mentally divide the prostate in 6 cores based on the perception from the TRUS images in real time and puncture each core 2 times to collect sample tissue for analysis, this division is illustrated in Figure 1.4.

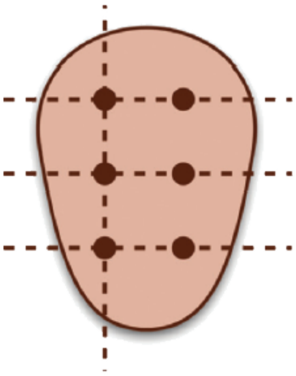


Figure 1.4: Sextant division of the prostate: coronal view of the core location. [8]

During post processing, the data collected was analysed in order to determine what was the real position and relative displacement of the prostate during each one of the perforations, knowing its initial position and volume, which is also affected during the exam. In Figure 1.5, one can observe the plotted results of one of the exams: In brown is the prostate model based on the images acquired at the beginning of the procedure, while the red points represent the position acquired at the time of each acquisition. The mean value of displacement at the end of all procedures was 17.96mm which is very relevant knowing that the normal diameter of a prostate is at maximum 40mm.

When confronted with these results, it became clear that the perception of the Urologist to where he thought the sample acquisitions were realised was very different from the real position of the needle in the prostate at the time of perforation.

One can conclude that the reliability on human perception for this exam can be hazardous for the

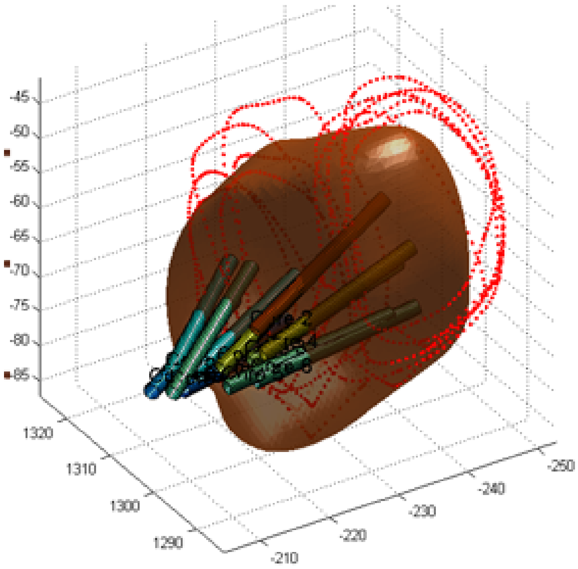


Figure 1.5: Prostate displacement and needle positions obtained from data analysis. [7]

patient. In many cases the doctor can miss the cancerous tissue which will lead to a false negative test. This represents a daily problem for urologists managing the disease, creates uncertainty and emotional stress for patients, and initiates a cascade of repeat testing and biopsies. There should be a better way with less painful implications and costs to the patient.

1.4 State Of The Art

This section will focus on the technological alternatives that have been developed to help improve the efficiency of the prostate biopsy. The main advance are physical mechanisms that allow the urologists to better guide the sample acquisition needle. One can divide these in two sections, one depending on the type of imaging used to locate the mechanism inside the patient, and another depending if the device is actuated or passive.

1.4.1 Passive Mechanisms

Beyersdorff et al. [9] developed and tested a passive device for transrectal biopsies. The system comprised a needle manipulated mechanically from outside a MRI machine. The mechanism can be seen in Figure 1.6. It is composed of a variable height stand on a base plate containing a track to enable the longitudinal axis movement of the system. Except for small parts such as screws, the device is made of synthetic materials that are fully compatible with MR imaging.

This device was tested in 12 patients who had one or more areas in the prostate that were suspicious for carcinoma at pre-biopsy MR imaging. Although the authors found their solution to be quite reliable to reach all the intern areas of the prostate, there was a clear limitation. It was necessary for the urologist to stand beside the MRI machine to manipulate the passive mechanism, guiding the needle inside the patient. This is a problem because it is not advisable for anyone to withstand the effects of radiation caused by the massive magnetic fields that extend beyond the machine.

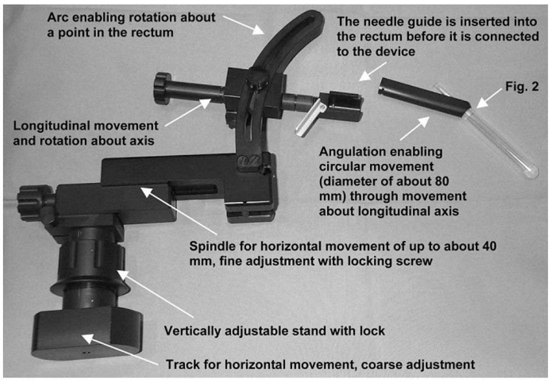

Arc enabling rotation about a point in the rectum
The needle guide is inserted into the rectum before it is connected to the device
Fig. 2
Longitudinal movement and rotation about axis
Angulation enabling circular movement (diameter of about 80 mm) through movement about longitudinal axis
Spindle for horizontal movement of up to about 40 mm, fine adjustment with locking screw
Vertically adjustable stand with lock
Track for horizontal movement, coarse adjustment

Figure 1.6: Passive mechanism with 2 linear tracks to be manipulated outside a MRI machine. [9]

Later, Bax et al. [10] created an initial 3D navigation system composed of a TRUS probe and a passive mechanical arm to minimize prostate motion. This system is shown in Figure 1.7. The validation of a prototype was done by testing on agar prostate phantoms. These tests showed that the needle guidance system had an accuracy mean error of 2.1 mm, and was able to diminish the prostate displacement caused by the needle to less than 1 mm.

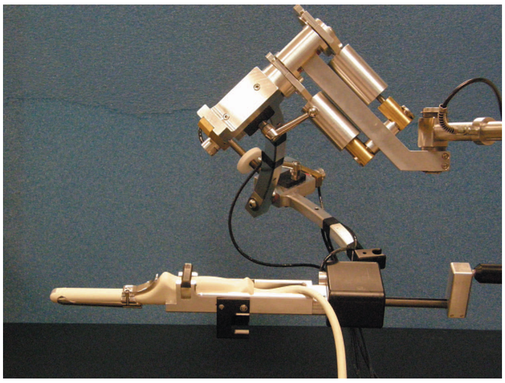


Figure 1.7: Passive mechanical arm with 3 DOFs with a TRUS probe attached. [10]

The previous mechanism was subsequently improved by Fenster et al. [11] to support any commercially available end-firing TRUS probe for prostate biopsy, and to be mounted at the base of a stabilizer. This new configuration constrained the TRUS probe's motion to three degrees of freedom (DOFs) and one degree of translation along the axis of the probe. The axis of the joints converge to a common point on the spherical assembly, the remote center of motion (RCM). This design minimizes targeting errors as the TRUS probe is constrained through this stationary point. The urologist's movements are scaled down throught the RCM, enabling a precision similar to that of a robotic assisted machine.

The remaining issue that none of the above mechanisms were able to solve was the human error. Although the precision to position the needle was enhanced in all of them, the remaining problem is the inability of the doctor to know where is the exact entry point on the prostate where the sample tissue collection is optimal for cancer diagnosis.

1.4.2 MRI Guided Mechanisms

After analysing the use of passive, human actuated mechanisms, it is now necessary to look at robotised structures that allow needle guiding systems to be accurately positioned. In this section, the type of image fusion used to locate the needle inside the patient will be MRI based scanning.

Chinzei et al. [12] started by introducing a MR compatible robot for surgery. It was designed to coexist and to be cooperative with a surgeon inside an intra-operative MR scanner as depicted in Figure 1.8.

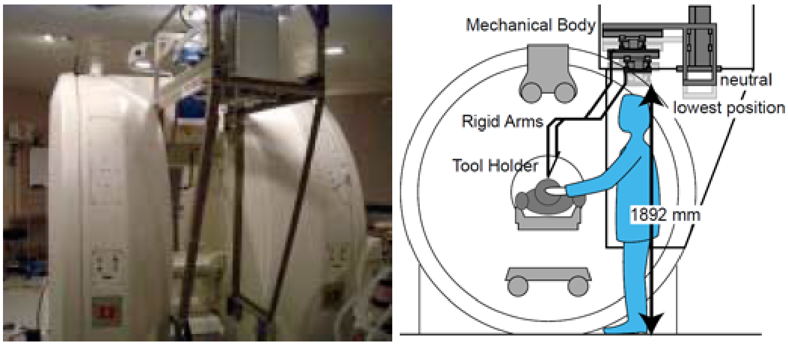


Figure 1.8: Intra-operative MRI robot for human-machine cooperation, with 2 arms and 5 DOFs in total. [12]

This type of MRI machine allows the surgeon to capture images of the patient while undergoing surgery. The robot is a parallel structure (closed loop), composed by two rigid arms, one with three linear DOFs and another with two. The arms are connected by a sliding bar and two spherical joints at the end of each arm, in a total of 5 DOFs. Although this was a very promising study of robot feasibility for prostate interventions, the main problem is the rarity of these machines worldwide.

Krieger et al. [13] developed an actuated manipulator arm for prostate biopsies under MRI guidance. The manipulator consists of a needle guide and rectal sheath, positioning and insertion stage, flexible shafts and a base mount as seen in Figure 1.9. It contains two MR coils in the needle shaft to be identified in the MRI images of the biopsy. A first-generation prototype was experimented in anesthetised canines which demonstrated a maximum needle displacement error of 2 mm.

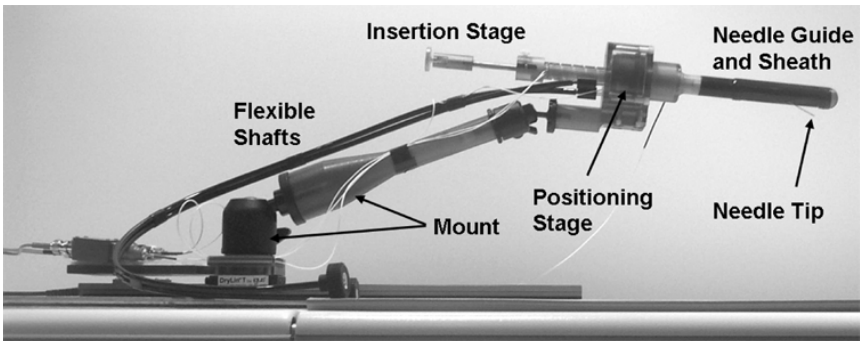


Figure 1.9: Actuated manipulator arm for MRI-guided biopsies with flexible shaft. [13]

Mozer et al. [14] made great advancements in this subject with the design of a robot that operates remotely alongside the patient inside a MRI machine as represented in Figure 1.10. This is the first mechanism here presented to be controlled remotely and not manually.

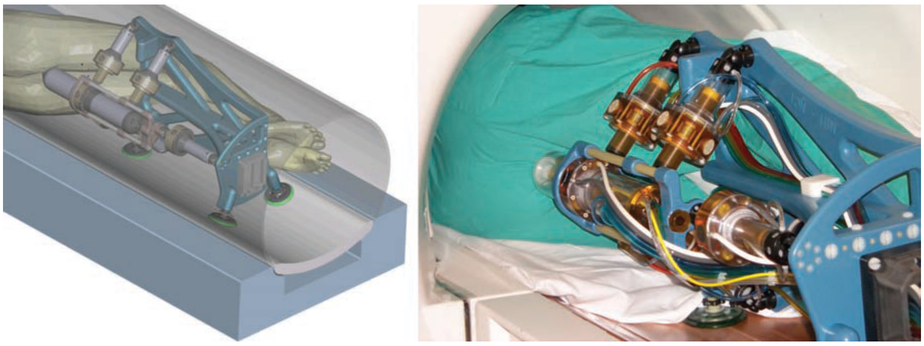


Figure 1.10: MrBot: Parallel robot with 5 MRI compatible actuators and 6 DOFs, for remote control. [14]

The robot is a parallel mechanism, with five linear actuators that connect the probe support to a base frame, and another linear actuator responsible for the axial movement of the biopsy probe. It has a total of 6 DOFs, three for rotation and three for translation. To overcome MRI incompatibilities of the common actuators, a new type of motor was designed, the PneuStep. This is a pneumatic motor using optical feedback. The robot was tested for brachytherapy, which is a form of radiotherapy where radioactive seeds are placed inside the prostate. The precision tests in phantoms displayed a mean placement error of 0.72 ± 0.36mm. These are very good results that stand on the edge of what is possible to do today.

Later, Yakar et al. [15] assessed the feasibility of using another pneumatically actuated, MRI remotely controlled device for biopsies of the prostate. However, this robotic system was a of serial architecture and was able to position the needle in three translational directions and two rotations, in a total of 5 DOFs. The setup for the experimental validation is shown in Figure 1.11.

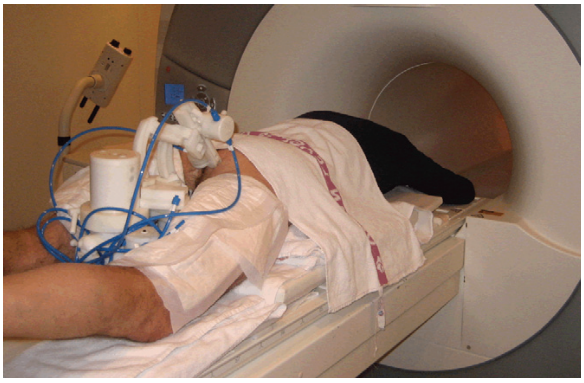


Figure 1.11: Rectal insertion of needle guide attached to MRI-compatible robot with 5 DOFs. [15]

The robot was able to reach 76% (13 out of 17) of the cancer-suspicious regions that were detected in 10 patients. One can conclude that it was due to this robot having less degrees of freedom than the previous one. Also, being a serial robot (open loop structure), the stability of the needle may be compromised with vibrations provoked by the wrong remote manipulation of the doctor (in general, a serial robot will be more susceptible to sudden accelerations than a parallel one).

Another relevant contribution by Su et al. [16] was the design and test of a more simple MRI-compatible actuated robot with piezoelectric actuation. This system was different in the fact that enabled the control of a curved, steerable active cannula. A prototype can be seen in Figure 1.12.

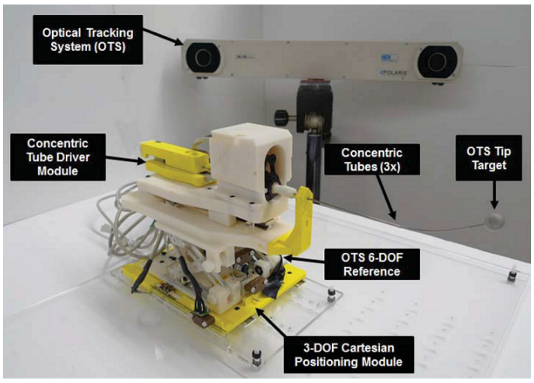

Optical Tracking System (OTS)
Concentric Tube Driver Module
Concentric Tubes (3x)
OTS Tip Target
OTS 6-DOF Reference
3-DOF Cartesian Positioning Module

Figure 1.12: Piezoelectrically actuated robot for MRI-guided brachytherapy with 6 DOFs. [16]

The robot consists of a modular 3 DOFs cannula driver with a fiducial tracking frame on a 3 DOFs atuated cartesian stage. In total it has 6 DOFs and a linear motion precision at joint-level of 0.03 mm. Preliminary phantom trials showed cannula trajectories with a maximum placement error of 2.24 mm.

From all of this devices, one can conclude that the approach for MRI-guided robotised structures is very effective. They help urologists have a better perception of where the needle is, during the biopsy.

However, there are some issues with this approach. The first is the necessity to develop specific equipment for MRI-based procedures that doesn't interfere with the magnetic fields. This may require additional investments and make the machines less commercially viable, which would result in less patients having access to them. Another issue may be the reduced comfort of the patient for having to spend a lot more time inside the MRI machine as it would be necessary for a normal scan, which can also augment the amount of radiation to what they are exposed.

The final and maybe most important issue one can find here is that we are completely removing the urologist from the physical scenario. By using totally actuated mechanisms, remotely controlled, the benefits of the human hand are lost. On one side, the machine will be more precise, but on the other, the human expertise and sensitivity to manoeuvre the probe is still necessary.

Also, usually, co-manipulation is a much more accepted resource for the medical community in the use of robotics in surgical environments.

1.4.3 TRUS Guided Mechanisms

In this section, actuated structures to improve the accuracy of the prostate biopsy are analysed. Only this time, only the TRUS-based methods will be considered.

Ho et al. [17] introduced the first robotic TRUS-guided prostate intervention. This robot is composed by three components: a needle gun holder, a TRUS probe holder and a gantry as represented in Figure 1.13. The gantry is the part responsible for the positioning in 2 axes while a linear motor controls the axial proximity of the probe. The robot is then placed on top of a spherical joint that allows its orientation to be fully defined, having a total of 6 DOFs. The security of the device was tested on phantoms, being verified in defined boundaries with a repeatability of less than 1mm.

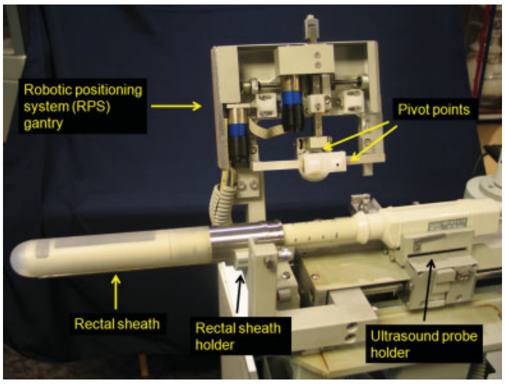


Figure 1.13: Gantry and TRUS holder of the TRUS-guided, 6 DOFs robot. [17]

The next robot is the only in this enumeration that consolidates all our previous recommendations for what the perfect device could be. Poquet et al. [18] developed a co-manipulator to assist urologists in prostate biopsies. This robot can function in two modes: a free mode, where the doctor can position the probe with no constraints; and a locked mode, where the robot maintains the probe at a specific location. The name of the robot is Apollo and it is a 6 DOFs robot, shown in Figure 1.14.

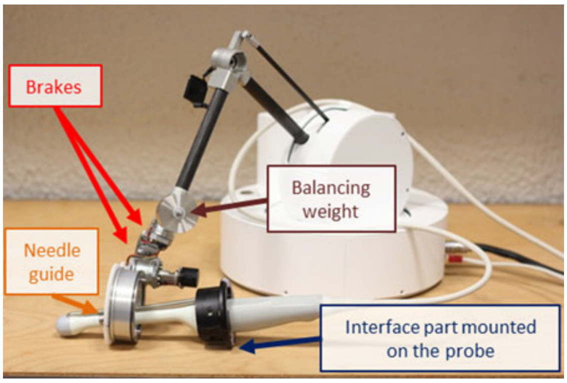


Figure 1.14: Apollo Robot for TRUS-guided biopsies with 6 DOFs. [18]

With an optical tracking system similar to the one used by Tavares [7], the authors tested the transparency of the robot in given trajectories. They used 12 people without experience in prostate needle placement to manipulate the robot and in-cadavero experiments to test the lock mode. The results showed that large errors occurred due to external forces when the wrist was not equipped with brakes and joint sensors.

1.5 Proposed Solution

From the various devices presented in the previous section, it is noticeable the advantage of cooperation between automated mechanisms and the knowledge of the human physicist. In this work, a step further is taken in human-robot interaction for medical applications. The development of a parallel robot is presented, to be implemented with a TRUS probe for prostate biopsies, which is to be manipulated by an urologist. The objective of the device is to guide the doctor to the optimal points of perforation through trajectory control, by providing real-time feedback from TRUS images. The diagram in Figure

1.15 intends to show the robot to be used and the general proposed setup.

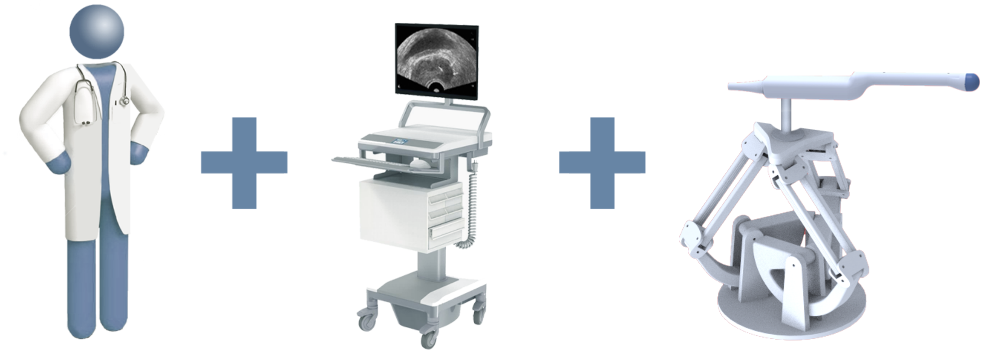


Figure 1.15: General schematic of the proposed solution.

As seen in the previously proposed approaches, serial robots lack stability and stiffness. Because they are series of successive masses connected to each other, a slight oscillation in the base will be transmitted to the consecutive links. The vibration on the end-effector will worsen as the number of links increases. The joints of the links closer to the base will have to support the total weight of the robot. This means that in dynamical terms, the lower actuators will need to offer greater torques than the ones closer to the end-effector.

These problems are the reason that in this work, a parallel robot was chosen. When dealing with prostate related medical interventions, it is very important to guarantee millimetric precision and patient safety. A parallel robot is a closed loop structure. This means that the position error of the end-effector caused by possible bending of the serial structure, is evened out by the addition of one or more legs connected to the base. In a way, one can see this as a structure that is connected to the ground in more than one points.

Because these robots are complex and usually expensive to build from scratch, a *Novint Falcon*^{TM} haptik controller acquired from Novint Technologies, Inc. is used as a prototype of the robot to be used. However, this device is disassembled and kinematically optimized to reach a proper workspace for the prostate biopsy test. Then, links with the new dimensions are designed and built for the prototype. The main advantage is that the power source and actuators remain the same as the original haptik controller.

For this robot to be able to reach all the points where the urologist would normally place the probe, some data was needed. Using the data provided by the research done in 2012 [7], a set of 15 model patients was chosen from the 22 originally registered. This registration was done using a *Polaris Spectra*® optical tracking system acquired from Northern Digital Inc. An optical marker compatible with the system was attached to the probe, and its position and orientation was captured during the prostate biopsies. With this, a workspace for the robot was inferred and its kinematic parameters optimized for the said space. A prototype was designed and built and afterwards, a control model is proposed for the parallel robot. Finally, all the implementation will be detailed, passing through the modelling, assembly and software used to achieve our proposed solution. A general outline of this work is represented in the diagram of Figure 1.16.

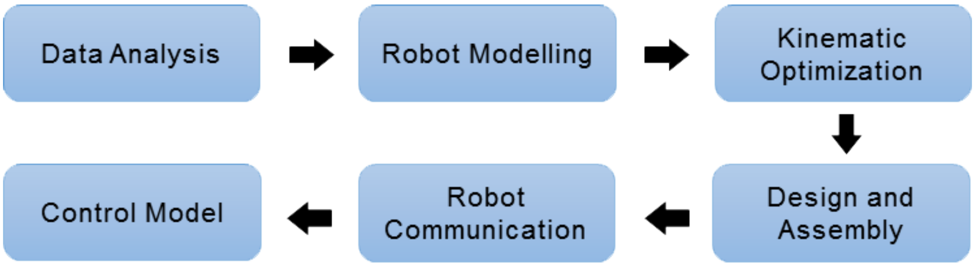


Figure 1.16: General diagram of the work phases.

1.6 Thesis Outline

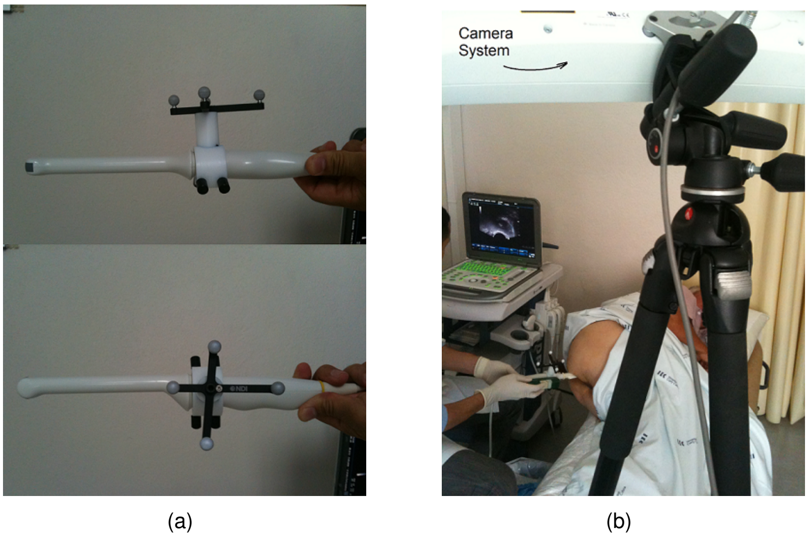
The developed work is presented in this thesis and divided into five general chapters. The first chapter, was a brief explanation of why this work was motivated, what kind of studies have been previously developed on the subject and what solution did the author commit to. Chapter 2 will describe how the data that was used in the optimization was previously acquired, and also, how this data was analysed and processed. This data is the collection of trajectories of a TRUS probe during prostate biopsies.

Afterwards, Chapter 3 details all about the robot physical description, kinematic and dynamic models. The following Chapter 4 is about the optimization of the kinematic parameters of the robot, and will go through the algorithm used and the results of the simulation. In Chapter 5 the whole implementation of the then previously presented theoretical results is detailed. Finally, in Chapter 6, the initial tests to control the Falcon Novint structure will be demonstrated, and a control model for the new robot will be proposed.

2 | Data Acquisition

2.1 Registration Procedure

This section will start by explaining how the data used in this thesis was acquired. The data is composed by a set of experimental results obtained by Tavares [7], during 22 transrectal prostate biopsy exams. The exams were anonymously recorded at Hospital Curry Cabral, in Lisbon.



(a) (b)

Figure 2.1: (a) Probe with the optical marker tool; (b) Acquisition setup for the biopsy procedure. [7]

The objective was to record the pose of the TRUS probe during each one of the exams. To do this, a *Polaris Spectra*® optical tracking system by Northern Digital Inc. was used. A marker tool compatible with this system was attached to the probe (as shown in Figure 2.1(a)), which allowed the *Polaris* to capture the location and orientation of the probe. This setup in shown in Figure 2.1(b).

2.2 Data Analysis

The raw data was delivered as '.mat' files and each one of the 22 patients had a different number of data acquisition files. The first step was to collect all these points and assemble a raw data cloud point for each one of the patients. Inside each acquisition file a structure called 'datapolaris' contained the registered data by the system, which was a character array of the type shown in Figure 2.2:

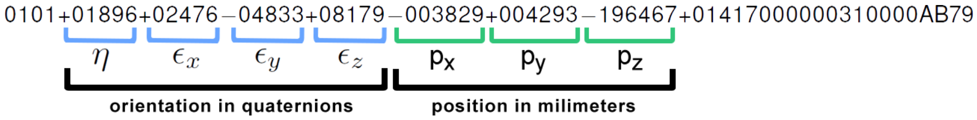
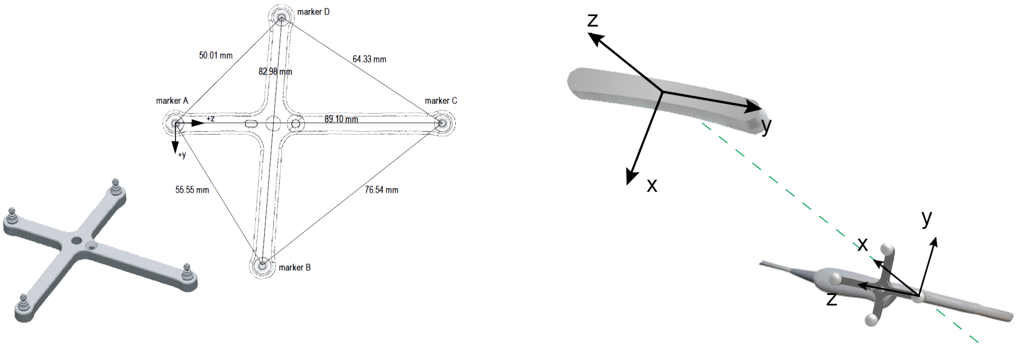

0101+01896+02476−04833+08179−003829+004293−196467+01417000000310000AB79
η
ϵ_x
ϵ_y
ϵ_z
p_x
p_y
p_z
orientation in quaternions
position in milimeters

Figure 2.2: Example of character array returned from *Polaris Spectra*®.

To understand the data retrieved, the 'Passive Polaris Spectra User Guide' [19] was consulted. The information acquired revealed that the orientation of the probe was returned in quaternions, displaying its four elements from character number 5 to number 29. The position of the probe was returned in milimiters from character number 30 to 51. Because this cloud point data is needed for the optimization of a kinematic structure, it is necessary to do some pre-processing. First, the original data comes in the referential of the *Polaris*, which is observing the position of the probe, or the origin of the reference frame of the marker as represented in Figure 2.3(b). Consulting the manual once again, one can know that the marker tool used has the reference frame shown in Figure 2.3(a).



(a)

(b)

Figure 2.3: (a) Marker tool used and respective diagram [19]; Polaris and marker tool reference frame.

It was necessary to transform all these points of data to the reference frame where our robot would be virtually placed. For this, an average point and orientation had to be computed in order to project the data cloud into this virtual plane.

2.2.1 Quaternions Average

To calculate the previously referred coordinate system, it was necessary to average the poses retrieved from the data. Knowing that the quaternion parameters are defined as $\mathbf{q} = [\eta, \epsilon_x, \epsilon_y, \epsilon_z]^T$ and can be calculated from a transformation matrix $\mathbf{T}$ such as:

$$\eta = \frac{1}{2}\sqrt{T_{1,1} + T_{2,2} + T_{3,3} + 1} \tag{2.1}$$

$$\epsilon_x = \frac{1}{2} sign(T_{3,2} - T_{2,3})\sqrt{T_{1,1} - T_{2,2} - T_{3,3} + 1} \tag{2.2}$$

$$\epsilon_y = \frac{1}{2} sign(T_{1,3} - T_{3,1})\sqrt{T_{2,2} - T_{1,1} - T_{3,3} + 1} \tag{2.3}$$

$$\epsilon_z = \frac{1}{2} sign(T_{2,1} - T_{1,2})\sqrt{T_{3,3} - T_{1,1} - T_{2,2} + 1} \tag{2.4}$$

To properly obtain the average of quaternion parameterized orientations, one needs to consider the Frobenius norm. Humbert et al. [20] proposed: Given two quaternions $\mathbf{q}$ and $\mathbf{q}_k$, on the unit sphere S_3 the Euclidean distance is defined by:

$$||\mathbf{q} - \mathbf{q}_k|| = ||\ \mathbf{I} - \mathbf{q}_k\mathbf{q}^{-1}|| = 2\sin\frac{\eta_k}{4} \tag{2.5}$$

The square of Euclidean distance is related to the scalar product of the quaternions such as:

$$||\mathbf{q} - \mathbf{q}_k||^2 = d_k^2 = 2(1 - \cos\frac{\eta_k}{2}) = 2[1 - (\eta\ \eta_k + \mathbf{q}.\mathbf{q}_k)] \tag{2.6}$$

For a given set of N rotations, the average orientation is obtained by minimizing the objective function described below.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{N}\sum_{k=1}^{N} d_k^2 = \sum_{k=1}^{N} 2[1 - (\eta\ \eta_k + \mathbf{q}.\mathbf{q}_k)] \\ \text{subject to} \quad & ||\mathbf{q}||^2 = \eta\ \eta + \mathbf{q}.\mathbf{q} = 1 \end{aligned} \tag{2.7}$$

Which, after some extensive Lagrange multipliers calculations follows that the mean orientation is given by the arithmetic average of the quaternions, normalized by the norm of the sum as detailed by the following equation.

$$\bar{\mathbf{q}} = \frac{(\mathbf{q}_1 + \mathbf{q}_2 + \cdots + \mathbf{q}_N)}{||\mathbf{q}_1 + \mathbf{q}_2 + \cdots + \mathbf{q}_N||} \tag{2.8}$$

Taking by example Figure 2.4, in this case, the average coordinate frame is at the cartesian center of the points, and its x0y plane is a top projection of the movements of the urologist. In order to use this information, one needs to know how to convert quaternions into a transformation matrix. This is done as presented in equation 2.9.

$$T(\epsilon_x, \epsilon_y, \epsilon_z, \eta) = \begin{bmatrix} 1 - 2(\epsilon_y^2 + \epsilon_z^2) & 2(\epsilon_x\epsilon_y - \epsilon_z\eta) & 2(\epsilon_x\epsilon_z + \epsilon_y\eta) & p_x \\ 2(\epsilon_x\epsilon_y + \epsilon_z\eta) & 1 - 2(\epsilon_x^2 + \epsilon_z^2) & 2(\epsilon_y\epsilon_z - \epsilon_x\eta) & p_y \\ 2(\epsilon_x\epsilon_z - \epsilon_y\eta) & 2(\epsilon_y\epsilon_z + \epsilon_x\eta) & 1 - 2(\epsilon_x^2 + \epsilon_y^2) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

Finally, for the cloud of points of all patients an average frame was obtained and can be used as the reference frame of origin for the robot to be designed.

2.2.2 Workspace Characterization

The workspace needed for the manipulation of the probe during the biopsies can be characterized from the data sets and the origin reference frame obtained. In Figure 2.4 and 2.5, it is plotted the points cloud of one representative patient and the average coordinate frame obtained.

The points registered for the probe are displayed such that the blue vectors indicate the longitudinal axis of the probe, and the green ones represent a perpendicular to the front side of the probe. The color scheme red, green, blue matches the x y z axis of the average frame, respectively. It is visible that the probe's displacement lies in an arched trajectory around the gluteus of the patient.

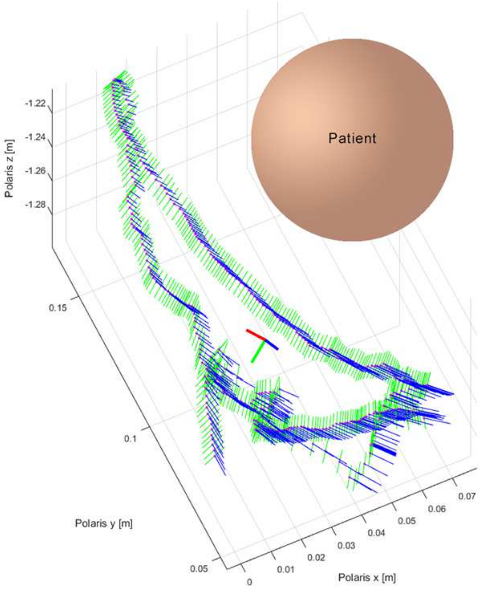


Figure 2.4: Points cloud of one patient and respective average coordinate frame - top view.

It is possible to measure the approximate cartesian coordinates needed to fulfil the trajectory of a regular prostate biopsy. In the side view of the points cloud, Figure 2.5, one can see that the range of positions in x varies from 0 to 0.07m, in the y axis varies from 0.05m to 0.15m and in z from -1.30m to -1.2m. This results in a displacement needed of around 7cm in x, 10 cm in y and 10 cm in the z axis. Also, the rotation of the probe along its point of fixation (inside the patient) should describe and arc of around 45º in the sagittal plane.

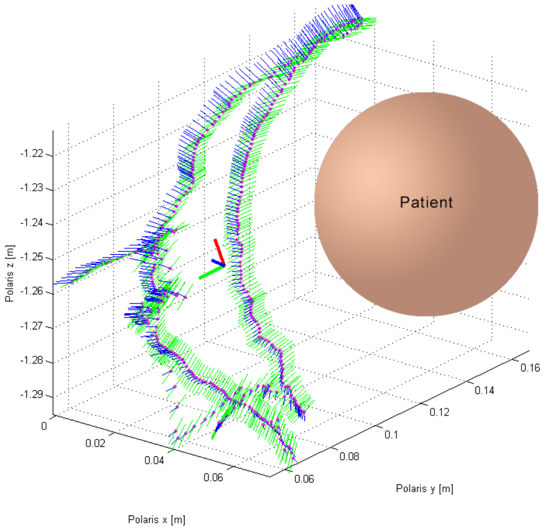


Figure 2.5: Points cloud of one patient and respective average coordinate frame - side view.

3 | Robot Modelling

A *Novint Falcon*^{TM} haptik controller acquired from Novint Technologies, Inc. as the one shown in Figure 3.1 is to be disassembled, kinematically optimized and reconstructed. For this it is necessary to model its structure.

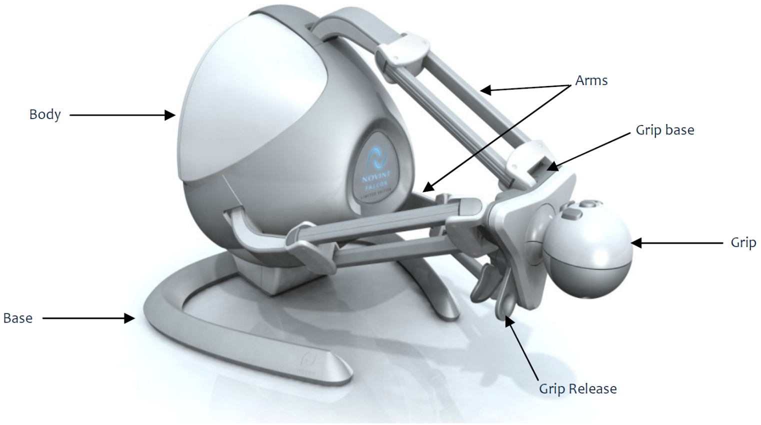


Figure 3.1: *Novint Falcon*^{TM} haptik controller acquired from Novint Technologies, Inc. [21]

A generic robot model is presented and is divided in four parts: the geometric model, which defines the cartesian coordinates $\mathbf{x}$ of the end effector of the robot as a function of the joint positions $\mathbf{q}$ in equation 3.1; the kinematic model, which characterizes the motion velocity of the end effector $\dot{\mathbf{x}}$ as related to the joint positions $\mathbf{q}$ by the kinematic jacobian $\mathbf{J}$ multiplied by the joint velocities $\dot{\mathbf{q}}$ in equation 3.2; the kinematic model of second order, which is the derivative of the previous one, resulting in the relation between the acceleration of the end effector $\ddot{\mathbf{x}}$ as a function of the joint positions $\mathbf{q}$, velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$, in equation 3.3; and lastly, the dynamic model which relates the cartesian forces at the end effector $\mathbf{F}$ with the inertia matrix $\mathbf{B}$, the coriolis and centrifugal terms matrix $\mathbf{C}$ and the gravitation matrix $\mathbf{G}$, in equation 3.4.

$$\mathbf{x} = L(\mathbf{q}) \tag{3.1}$$

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{3.2}$$

$$\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \tag{3.3}$$

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \tau = \mathbf{J}^T\mathbf{F} \tag{3.4}$$

3.1 Geometric Model

In this section, the geometric model from equation 3.1 will be deduced in the inverse form. This demonstration was based on the approaches of modelling done by Stamper [22] for a simpler 3 DOF parallel manipulator and Shah et al. [23]. A diagram of the geometry of one of the legs of the Falcon is shown in Figure 3.2.

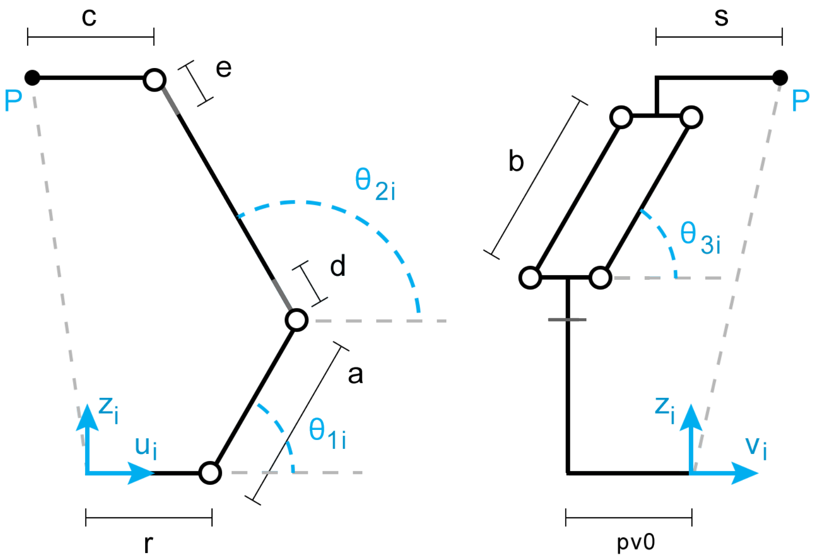


Figure 3.2: Representation of ith falcon leg on the left and the front view on the right.

In equations 3.5 to 3.7 the position of point $\mathbf{P}_{uvw} = [p_{ui}\ p_{vi}\ p_{wi}]^T$, which is the base of the end effector, is described in the UVW coordinate frame of the ith leg of the Falcon.

$$p_{ui} = a\cos(\theta_{1i}) - c + \cos(\theta_{2i})\ (2d + b\sin(\theta_{3i})) \tag{3.5}$$

$$p_{vi} = b\cos(\theta_{3i}) + s \tag{3.6}$$

$$p_{wi} = a\sin(\theta_{1i}) + \sin(\theta_{2i})\ (2d + b\sin(\theta_{3i})) \tag{3.7}$$

It is also needed to consider the positions of each leg and orientation of each one of their reference coordinate frames UVW, in the robot global frame XYZ. A diagram of this geometry is represented in Figure 3.3. A transformation of the point $\mathbf{P}_{xyz} = [p_x\ p_y\ p_z]^T$ on the global reference frame to each of the legs' coordinate system can be written as:

$$\begin{bmatrix} p_{ui} \\ p_{vi} \\ p_{wi} \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) & 0 \\ -\sin(\phi_i) & \cos(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} -r \\ p_{v0} \\ 0 \end{bmatrix} \tag{3.8}$$

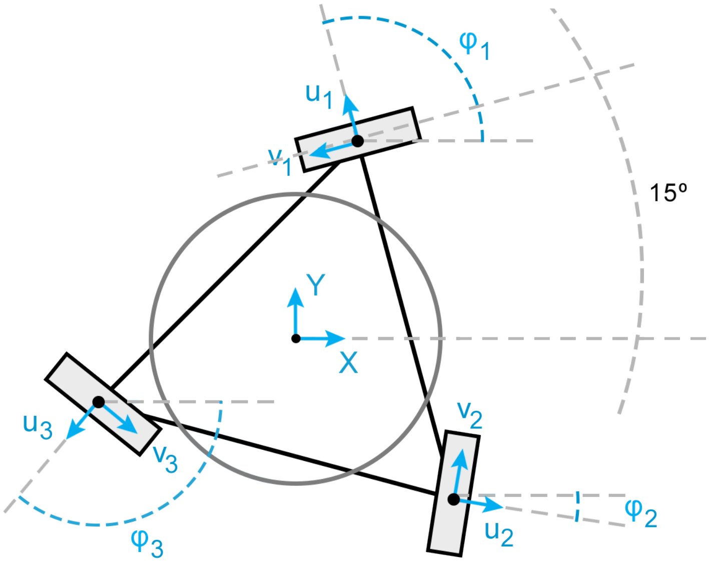


Figure 3.3: Representation of the base geometry of the falcon and position of each leg.

3.1.1 Inverse Geometric Model

With the information of the geometric parameters and the position in the robot's global cartesian space of the end effector, the solution for the third joint angle of the ith θ_{3i} can be found as described in expression 3.9.

$$\theta_{3i} = \arccos(\frac{p_{vi} - s}{b}) \tag{3.9}$$

Isolating $cos(\theta_{2i})$ in equation 3.5 and $sin(\theta_{2i})$ in equation 3.7 and then summing the squares of the two resultant equations, θ_{2i} is eliminated by the Pythagorean trigonometric identity, resulting that:

$$(p_{ui}+c)^2+p_{wi}^2+a^2-2a(p_{ui}+c)\cos(\theta_{1i})-2ap_{wi}\sin(\theta_{1i}) = (d+e)^2+2(d+e)b\sin(\theta_{3i})+b^2\sin^2(\theta_{3i}) \tag{3.10}$$

Defining an auxiliary half-angle tangent, the following relationships can be written.

$$t_{1i} = \tan(\frac{\theta_{1i}}{2}) \quad \leftrightarrow \quad \begin{cases} \sin(\theta_{1i}) = \frac{2t_{1i}}{1+t_{1i}^2} \\ \cos(\theta_{1i}) = \frac{1-t_{1i}^2}{1+t_{1i}^2} \end{cases} \tag{3.11}$$

Substituting the previous relations in equation 3.10, a simplified polynomial form is obtained.

$$l_{2i}t_{1i}^2 + l_{1i}t_{1i} + l_{0i} = 0, \quad \text{where} \tag{3.12}$$

$$\begin{cases} l_{0i} &= p_{wi}^2 + p_{ui}^2 + 2cp_{ui} - 4d^2 - b^2(\sin(\theta_{3i}))^2 - 4bd\sin(\theta_{3i}) - 2ap_{ui} + (a-c)^2 \\ l_{1i} &= -4ap_{wi} \\ l_{2i} &= p_{wi}^2 + p_{wi}^2 + 2cp_{ui} - 4d^2 - b^2(\sin(\theta_{3i}))^2 - 4bd\sin(\theta_{3i}) + 2ap_{ui} + (a+c)^2 \end{cases}$$

A solution for θ_{1i} can be found from 3.12 with data from the end effector position. Considering the only form that is applicable to the Falcon due to the range of the angle, one obtains:

$$\theta_{1i} = 2\arctan(t_{1i}), \quad \text{where} \quad t_{1i} = \frac{-l_{1i} - \sqrt{(l_{1i}^2 - 4l_{2i}l_{0i})}}{2l_{2i}} \tag{3.13}$$

Knowing the value of θ_{1i} and combining 3.5 and 3.7 , one can write θ_{2i} such that:

$$\theta_{2i} = \arctan\Big(\frac{p_{wi} - a\sin(\theta_{1i})}{p_{ui} - a\cos(\theta_{1i}) + c}\Big) \tag{3.14}$$

The inverse geometric model is fully defined by having the expressions for every joint angle as a function of the geometric parameters and the end effector cartesian position.

3.2 Kinematic Model

In this section, the kinematic model from equation 3.2 will be deduced in the inverse form. To compute the kinematic model, a geometric jacobian $\mathbf{J}$ is to be defined. Equations 3.5 to 3.7 will be differentiated with respect to time in equations 3.15 to 3.17, which represent the velocity of point $\dot{\mathbf{P}}_{uvw} = [\dot{p}_{ui}\;\dot{p}_{vi}\;\dot{p}_{wi}]^T$, in each leg's coordinate system.

$$\dot{p}_{ui} = -\Big(a\sin(\theta_{1i})\Big)\dot{\theta}_{1i} - \Big(\sin(\theta_{2i})\;(2d + b\sin(\theta_{3i}))\Big)\dot{\theta}_{2i} + \Big(b\cos(\theta_{2i})\cos(\theta_{3i})\Big)\dot{\theta}_{3i} \tag{3.15}$$

$$\dot{p}_{vi} = -\Big(b\sin(\theta_{3i})\Big)\dot{\theta}_{3i} \tag{3.16}$$

$$\dot{p}_{wi} = \Big(a\cos(\theta_{1i})\Big)\dot{\theta}_{1i} + \Big(\cos(\theta_{2i})\;(2d + b\sin(\theta_{3i}))\Big)\dot{\theta}_{2i} + \Big(b\sin(\theta_{2i})\cos(\theta_{3i})\Big)\dot{\theta}_{3i} \tag{3.17}$$

Solving equation 3.16 for $\dot{\theta}_{3i}$, one obtains:

$$\dot{\theta}_{3i} = -\frac{\dot{p}_{vi}}{b\sin(\theta_{3i})} \tag{3.18}$$

Substituting equation 3.18 in equations 3.15 and 3.17, it is possible to remove $\dot{\theta}_{3i}$ as an unknown, remaining $\dot{\theta}_{1i}$ and $\dot{\theta}_{2i}$. To eliminate $\dot{\theta}_{2i}$, equations 3.19 and 3.20 are combined in equation 3.21.

$$\begin{aligned}\dot{p}_{ui}\sin(\theta_{3i}) = &-\Big(\cos(\theta_{2i})\cos(\theta_{3i})\Big)\dot{p}_{vi} - \Big(a\sin(\theta_{1i})\sin(\theta_{3i})\Big)\dot{\theta}_{1i} \\ &-\Big(\sin(\theta_{2i})\sin(\theta_{3i})\ (2d + b\sin(\theta_{3i}))\Big)\dot{\theta}_{2i} \quad \Leftrightarrow\end{aligned}$$

$$\Leftrightarrow \quad \dot{\theta}_{2i} = \frac{-\dot{p}_{ui}\sin(\theta_{3i}) - \Big(a\sin(\theta_{1i})\sin(\theta_{3i})\Big)\dot{\theta}_{1i} - \Big(\cos(\theta_{2i})\cos(\theta_{3i})\Big)\dot{p}_{vi}}{\sin(\theta_{2i})\sin(\theta_{3i})\ (2d + b\sin(\theta_{3i}))} \tag{3.19}$$

$$\begin{aligned}\dot{p}_{wi}\sin(\theta_{3i}) = &-\Big(\sin(\theta_{2i})\cos(\theta_{3i})\Big)\dot{p}_{vi} + \Big(a\cos(\theta_{1i})\sin(\theta_{3i})\Big)\dot{\theta}_{1i} \\ &+\Big(\cos(\theta_{2i})\sin(\theta_{3i})\ (2d + b\sin(\theta_{3i}))\Big)\dot{\theta}_{2i} \quad \Leftrightarrow\end{aligned}$$

$$\Leftrightarrow \quad \dot{\theta}_{2i} = \frac{\dot{p}_{wi}\sin(\theta_{3i}) - \Big(a\cos(\theta_{1i})\sin(\theta_{3i})\Big)\dot{\theta}_{1i} + \Big(\sin(\theta_{2i})\cos(\theta_{3i})\Big)\dot{p}_{vi}}{\cos(\theta_{2i})\sin(\theta_{3i})\ (2d + b\sin(\theta_{3i}))} \tag{3.20}$$

$$\begin{aligned}-\Big(\sin(\theta_{3i})\cos(\theta_{2i})\Big)\dot{p}_{ui} - \Big(a\sin(\theta_{1i})\sin(\theta_{3i})\cos(\theta_{2i})\Big)\dot{\theta}_{1i} - \Big(\cos^2(\theta_{2i})\cos(\theta_{3i})\Big)\dot{p}_{vi} = \\ \Big(\sin(\theta_{3i})\sin(\theta_{2i})\Big)\dot{p}_{wi} - \Big(a\cos(\theta_{1i})\sin(\theta_{3i})\sin(\theta_{2i})\Big)\dot{\theta}_{1i} + \Big(\sin^2(\theta_{2i})\cos(\theta_{3i})\Big)\dot{p}_{vi} \quad \Leftrightarrow\end{aligned}$$

$$\Leftrightarrow \dot{p}_{ui}\Big(\sin(\theta_{3i})\cos(\theta_{2i})\Big) + \dot{p}_{vi}\Big(\cos(\theta_{3i})\Big) + \dot{p}_{wi}\Big(\sin(\theta_{3i})\sin(\theta_{2i})\Big) + \dot{\theta}_{1i}\Big(a\sin(\theta_{1i} - \theta_{2i})\sin(\theta_{3i})\Big) \tag{3.21}$$

Equation 3.21 will be referred to as f_i from here onwards. Once again, it is needed to consider the transformation of the velocity in each leg to the global reference frame XYZ. Differentiating 3.8 with respect to time, one obtains 3.22 as a function of the velocity of point $\dot{\mathbf{P}}_{xyz} = [\dot{p}_x\ \dot{p}_y\ \dot{p}_z]^T$.

$$\begin{bmatrix}\dot{p}_{ui}\\ \dot{p}_{vi}\\ \dot{p}_{wi}\end{bmatrix} = \begin{bmatrix}\cos(\phi_i) & \sin(\phi_i) & 0\\ -\sin(\phi_i) & \cos(\phi_i) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{p}_x\\ \dot{p}_y\\ \dot{p}_z\end{bmatrix} \Leftrightarrow \dot{\mathbf{P}}_{uvw} = \mathbf{T}_{xyz}^{uvw}\ \dot{\mathbf{P}}_{xyz} \tag{3.22}$$

3.2.1 Geometric Jacobian

To complete the kinematic model, it is necessary to define the geometric jacobian of the structure. Computing the partial derivatives of Equation f_i (3.21) with respect to $\dot{\mathbf{P}}_{uvw}$ and $\dot{\theta}_{\mathbf{1}}$, respectively in equation 3.25 and 3.26 one obtains $\mathbf{A}r_{uvw}$ and $\mathbf{B}$. It is also necessary to transform the position to the XYZ cartesian global reference frame by 3.22. Finally, one obtains the relation between actuated joint velocities $\dot{\theta}_{\mathbf{1}} = [\dot{\theta}_{11}\ \dot{\theta}_{12}\ \dot{\theta}_{13}]^T$ and end effector velocity $\dot{\mathbf{P}}_{xyz}$.

$$\mathbf{A}r_{uvw}\mathbf{T}_{xyz}^{uvw}\ \dot{\mathbf{P}}_{xyz} = \mathbf{B}\ \dot{\theta}_{\mathbf{1}} \tag{3.23}$$

Thus, the geometric jacobian can be defined by 3.24.

$$\mathbf{J} = (\ \mathbf{A}r_{uvw}\mathbf{T}_{xyz}^{uvw}\ )^{-1}\mathbf{B} \quad \Leftrightarrow \quad \mathbf{J} = (\mathbf{A}_r)^{-1}\mathbf{B} \tag{3.24}$$

Where, the partial derivatives are computed as follows:

$$\mathbf{A}r_{uvw} = \begin{bmatrix} \frac{\partial f_1}{\partial \dot{p}_{u,1}} & \frac{\partial f_1}{\partial \dot{p}_{v,1}} & \frac{\partial f_1}{\partial \dot{p}_{w,1}} \\ \frac{\partial f_2}{\partial \dot{p}_{u,2}} & \frac{\partial f_2}{\partial \dot{p}_{v,2}} & \frac{\partial f_2}{\partial \dot{p}_{w,2}} \\ \frac{\partial f_3}{\partial \dot{p}_{u,3}} & \frac{\partial f_3}{\partial \dot{p}_{v,3}} & \frac{\partial f_3}{\partial \dot{p}_{w,3}} \end{bmatrix} \tag{3.25}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial \dot{\theta}_{11}} & 0 & 0 \\ 0 & \frac{\partial f_2}{\partial \dot{\theta}_{12}} & 0 \\ 0 & 0 & \frac{\partial f_3}{\partial \dot{\theta}_{13}} \end{bmatrix} \tag{3.26}$$

Obtaining $\mathbf{A}_r$ and $\mathbf{B}$ in Equations 3.27 and 3.28.

$$\mathbf{A}_r = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \tag{3.27}$$

$$\mathbf{B} = \begin{bmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_3 \end{bmatrix} \tag{3.28}$$

Where

$$A_{i1} = \cos(\theta_{2i}) \sin(\theta_{3i}) \cos(\phi_i) - \cos(\theta_{3i}) \sin(\phi_i) \tag{3.29}$$

$$A_{i2} = \cos(\theta_{2i}) \sin(\theta_{3i}) \sin(\phi_i) + \cos(\theta_{3i}) \cos(\phi_i) \tag{3.30}$$

$$A_{i3} = \sin(\theta_{2i}) \sin(\theta_{3i}) \tag{3.31}$$

$$B_i = a \sin(\theta_{2i} - \theta_{1i}) \sin(\theta_{3i}) \tag{3.32}$$

3.2.2 Inverse Kinematic Model

If the geometric parameters and the position and acceleration of the end effector in the robot's global frame are known, the solution for the joint angle accelerations of the active joints $\dot{\boldsymbol{\theta}}_{\mathbf{1}} = [\dot{\theta}_{11}\ \dot{\theta}_{12}\ \dot{\theta}_{13}]^T$ can be found as a reduced Inverse Kinematic Model in Equation 3.33. The term reduced refers to only the active or motorized joints being computed.

$$\mathbf{A}r_{uvw}\mathbf{T}^{uvw}_{xyz}\ \dot{\mathbf{P}}_{xyz} = \mathbf{B}\ \dot{\boldsymbol{\theta}}_{\mathbf{1}} \quad \Leftrightarrow \quad \dot{\boldsymbol{\theta}}_{\mathbf{1}} = \mathbf{J}_{inv}\ \dot{\mathbf{P}}_{xyz}$$

$$\mathbf{J}_{inv} = (\mathbf{B})^{-1}\mathbf{A}_r \tag{3.33}$$

3.3 Kinematic Model of Second Order

In this section, the kinematic model from equation 3.3 will be deduced. From section 3.2 it is already known the formulation for the relation between actuated joint velocities and end effector velocity. Here, one wishes to know the relation between actuated joint accelerations and the end effector cartesian acceleration. Taking Equation 3.33 and substituting the geometric jacobian from 3.24:

$$\mathbf{A}r\ \dot{\mathbf{P}}_{xyz} = \mathbf{B}\ \dot{\boldsymbol{\theta}}_{\mathbf{1}} \quad \Leftrightarrow \quad \dot{\mathbf{P}}_{xyz} = \mathbf{J}\ \dot{\boldsymbol{\theta}}_{\mathbf{1}} \tag{3.34}$$

It is necessary to differentiate this forward kinematic equation with respect to time, such that:

$$\ddot{\mathbf{P}}_{xyz} = \mathbf{J}\,\ddot{\theta}_{\mathbf{1}} + \dot{\mathbf{J}}\,\dot{\theta}_{\mathbf{1}} \tag{3.35}$$

The demonstration of how the final expression for the differentiation of the jacobian was obtained is detailed in Annex A. The computation of $\dot{\mathbf{J}}$ is detailed in Equation 3.36.

$$\dot{\mathbf{J}} = \frac{\mathbf{B}}{\det^2(\mathbf{A_r})}\left(\frac{d}{dt}adj(\mathbf{A_r})\det(\mathbf{A_r}) - adj(\mathbf{A_r})\frac{d}{dt}\det(\mathbf{A_r})\right) + \frac{\dot{\mathbf{B}}}{\det(A_r)}adj(A_r) \tag{3.36}$$

Where $adj(A_r)$ is the adjoint matrix of A_r and $\det(A_r)$ its determinant.

3.4 Dynamic Model

In this section, the dynamical model from equation 3.4 will be deduced. First, it is necessary to consider which forces are applied in each of the legs of the robot. The free body diagram for the ith leg can be seen in Figure 3.4. W represents the weight of the end effector platform while h_p is the torque in the first joint caused by the forces applied at the end effector P.

Two assumptions must be defined in terms of distribution of masses and gravity considerations:

1. The base of the robot is on the ground such that gravity acts in the negative global Z direction.

2. The mass m_b of each rod, in link b, is evenly divided and concentrated at joints B_i and E_i.

The last item can be assumed since the rods don't play a dominant role in the robot dynamics. This is because they are lighter than the rest of the manipulator and don't carry any of the load associated with the mass of the actuators. With this, one can write two equations for the sum of moments at the first joint. Equation 3.37 will define the balance of the inertia moments while Equation 3.38 details the resultant moment at A due to the motor torque τ_i and gravitational force g.

$$\sum M_{A_i} = I_A\ddot{\theta}_{1i} + c_d\dot{\theta}_{1i} + h_p \tag{3.37}$$

$$\sum M_{A_i} = \tau_i - ag\cos(\theta_{1i})(\frac{1}{2}m_a + m_b) - \mathbf{J}^T m_p\mathbf{g} \tag{3.38}$$

In the first equation I_A is the mass moment of inertia of the link a, accounting for the motor rotor inertia I_m and one of the rod masses m_b concentrated at B_i. h_p is the load at joint A_i due to the forces applied at P, considering the other rod mass concentrated at joint E_i. c_d is the viscous damping coefficient of the actuator. In the second equation, τ_i is the torque applied by the actuator and g is the gravity acceleration.

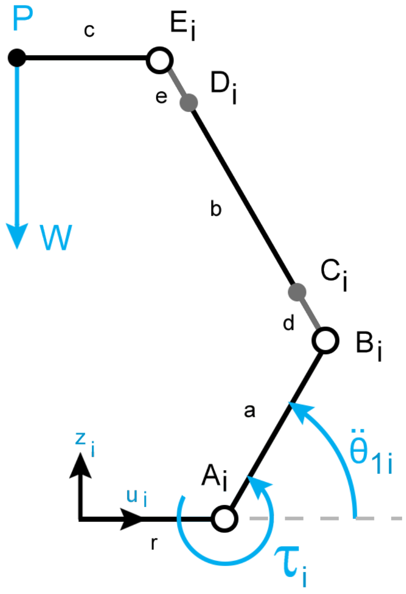


Figure 3.4: Representation of free body diagram for the ith falcon leg.

$$I_A = I_m + a^2(\frac{1}{3}m_a + m_b) \tag{3.39}$$

$$h_p = \mathbf{J}^T m_p \ddot{\mathbf{p}} \tag{3.40}$$

$$m_p = 3m_b + m_c \tag{3.41}$$

$$\mathbf{g} = [0 \quad 0 \quad g]^T \tag{3.42}$$

$Where,$

$g \Rightarrow$ Gravitational acceleration;

$I_m \Rightarrow$ Mass moment of inertia of the motor rotor;

$m_a \Rightarrow$ Mass of link a;

$m_b \Rightarrow$ Mass of each rod of link b;

$m_c \Rightarrow$ Mass of the end effector platform and payload;

$\mathbf{J} \Rightarrow$ Jacobian matrix from Equation 3.24;

$\ddot{\mathbf{p}} \Rightarrow$ Acceleration of the end effector in global reference frame.

From Equation 3.35 the acceleration of point P is related to the acceleration of the actuated joints. After combining Equation 3.37 and 3.38 for each leg, Equations 3.39 to 3.42 and 3.35 are substituted in the final matricial expression for all actuated joints.

$$I_A\ddot{\theta}_1 + c_d\dot{\theta}_1 + \mathbf{J}^T m_p\ddot{\mathbf{p}} = \tau_i + ag\big[\cos(\theta_{\mathbf{1}})\big]\big(\frac{1}{2}m_a + m_b\big) + \mathbf{J}^T m_p\mathbf{g} \quad \Leftrightarrow$$

$$I_A\ddot{\theta}_1 + c_d\dot{\theta}_1 + \mathbf{J}^T m_p\big(\mathbf{J}\ddot{\theta}_1 + \dot{\mathbf{J}}\dot{\theta}_1\big) = \tau_i + ag\big[\cos(\theta_{\mathbf{1}})\big]\big(\frac{1}{2}m_a + m_b\big) + \mathbf{J}^T m_p\mathbf{g} \quad \Leftrightarrow$$

$$\tau = \mathbf{M}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta})\dot{\theta} + \mathbf{G}(\theta) \tag{3.43}$$

Where $\tau = [\tau_1\ \tau_2\ \tau_3]^T$, $\mathbf{I}$ is the identity matrix and,

$$\mathbf{M}(\theta) = \left(I_m + a^2(\frac{1}{3}m_a + m_b)\right)\mathbf{I} + \mathbf{J}^T\left(3m_b + m_c\right)\mathbf{J} \tag{3.44}$$

$$\mathbf{C}(\theta, \dot{\theta}) = c_d\,\mathbf{I} + \mathbf{J}^T\left(3m_b + m_c\right)\dot{\mathbf{J}} \tag{3.45}$$

$$\mathbf{G}(\theta) = ag(\frac{1}{2}m_a + m_b)\begin{bmatrix}\cos(\theta_{11})\\ \cos(\theta_{12})\\ \cos(\theta_{13})\end{bmatrix} + \mathbf{J}^T\left(3m_b + m_c\right)\begin{bmatrix}0\\0\\g\end{bmatrix} \tag{3.46}$$

4 | Kinematic Optimization

The goal of this section is to detail the process of optimizing the kinematic structure of the robot described in Chapter 3. The criteria of optimization will be the maximum manipulability and number of points reached from the cloud point data retrieved as explained in Chapter 2.

4.1 Manipulability

The manipulability of a robot is an index to evaluate its performance. It is defined by the capacity of changing the position and orientation of the end effector given a joint configuration. This capability is described by the velocity manipulability ellipsoid [24] represented in Figure 4.1. Considering the euclidean norm of joint velocities

$$||\dot{\mathbf{q}}|| = \sqrt{\dot{q}_1^2 + \dot{q}_2^2 + \cdots + \dot{q}_n^2} \quad \text{such that} \quad ||\dot{\mathbf{q}}|| \leq 1, \tag{4.1}$$

the set of robot velocities $\dot{\mathbf{x}}$ which is realized by equation 4.1, is the manipulability ellipsoid in the euclidean space of dimension m, equal to the degrees of freedom of the robot [25]. The principal axes' directions of the manipulability ellipsoid will be the eigenvalues λ_i of $\mathbf{J}\mathbf{J}^T$, while their dimensions will be the singular values $\sigma_i = \sqrt{\lambda_i}$.

A global representative measure of manipulability can be obtained by considering the volume of the ellipsoid. This volume is proportional to $w(q)$, detailed in equation 4.2.

$$w(q) = \sqrt{\det(J\ J^T)} \tag{4.2}$$

The set of joint velocities of unitary norm, correspond to the perfect manipulability robot configuration $\dot{\mathbf{x}}$, as demonstrated below.

$$\dot{\mathbf{q}}^T \dot{\mathbf{q}} = 1 \Leftrightarrow \dot{\mathbf{x}}^T (J^{-T} J)\ \dot{\mathbf{x}} = 1 \Leftrightarrow \dot{\mathbf{x}}^T (J\ J^T)^{-1} \dot{\mathbf{x}} = 1 \tag{4.3}$$

At a singularity, the kinematic jacobian becomes rank deficient and its determinant null, such that $w(q) = 0$. So, this measure is also an indicator of the manipulator's proximity to a singular configuration.

It is also important to consider that there is a duality velocities/forces for the manipulability ellipsoid. This means that the same efforts of the actuators to move the end effector in a certain direction, given a joint configuration, can be translated to the efforts to apply a cartesian force in the same direction.

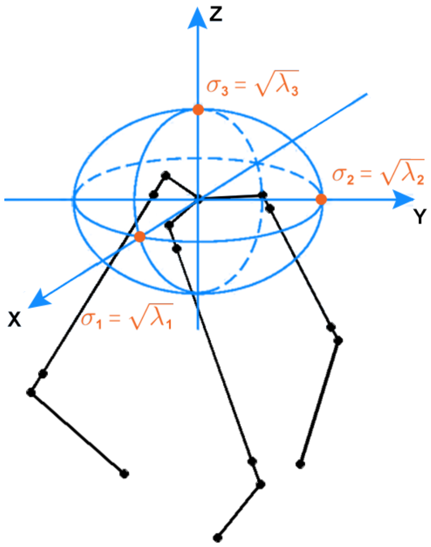

Z
$\sigma_3 = \sqrt{\lambda_3}$
$\sigma_2 = \sqrt{\lambda_2}$
Y
X
$\sigma_1 = \sqrt{\lambda_1}$

Figure 4.1: Manipulability ellipsoid representation on the kinematic structure of the *Novint Falcon*^{TM}, with its principal axes highlighted.

4.2 Parameters

In order to maximize the manipulability and number of points reached by the robot, it is necessary to use the geometric and kinematic model to test different geometric parameters. The original geometric parameters of the *Novint Falcon*^{TM} are specified in Table 4.1 [26].

Parameter	Value	Description
a [mm]	60	Distance from leg base to the first joint.
b[mm]	103	Length of the parallelogram link.
d=e [mm]	12	Length of link from first joint to parallelogram joint.
r [mm]	37	Distance from fixed frame origin to leg base, u component
pv0 [mm]	23	Distance from fixed frame origin to leg base, v component
c [mm]	16	Length from last joint connection point to end effector center, u component
s [mm]	-25	Length from last joint connection point to end effector centerr, v component
ϕ_1 [rad]	$\frac{7\pi}{12}$	Angle of first leg's coordinate system.
ϕ_2 [rad]	$\frac{-\pi}{12}$	Angle of second leg's coordinate system.
ϕ_3 [rad]	$\frac{-9\pi}{12}$	Angle of third leg's coordinate system.

Table 4.1: Geometric parameters of the *Novint Falcon*^{TM} as represented in Figure 3.2. [26]

These parameters are illustrated in Figure 3.2 and were confirmed by measuring the parts both physically and in the Computed Assisted Drawing (CAD) model. The CAD model was kindly provided by Mr. Steven Martin [27], Research Engineer at Australian Centre for Robotic Vision, Queensland

University of Technology.

After some analysis of what would be the best parameters to optimize, it was noticeable that there was few relevance of changing parameters c, s, ϕ_1, ϕ_2 and ϕ_3. This can be deduced after considering that it is desirable to keep the original end effector of the structure, for manufacturing simplicity. Also, the orientation of the coordinate frames of each leg was kept such as to maintain the overall symmetry of the structure. This being established, the parameters to optimize are declared in expression 4.4.

$$\text{Parameters} = [a,\ b,\ d,\ r,\ pv0,\ z] \tag{4.4}$$

Parameter z is to be defined as an auxiliary value. This parameter is created in the hope of increasing the number of points reached by the robot by moving the points cloud in the direction of the Oz axis. As explained in Chapter 2, the base reference frame of the robot will be assigned to the average frame computed from the points cloud average position and orientation. However, the workspace of this kinematic structure doesn't contain the origin of the base referential. Actually, the first reachable points are a few centimetres apart from the base reference frame in the positive direction of Oz axis. This explains the need for parameter z to move the points cloud as illustrated in Figure 4.2.

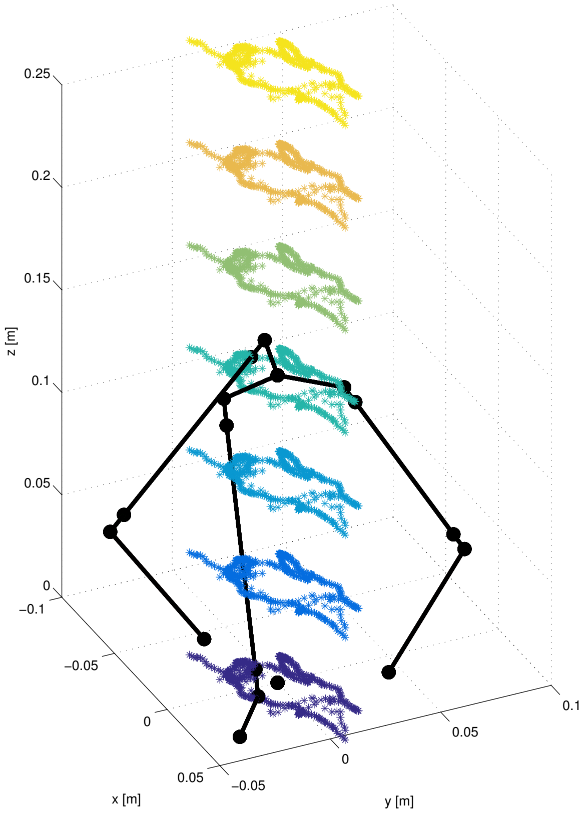


Figure 4.2: Illustration of the effect of increasing optimization parameter z, as to adequate the robot's reach in the points cloud. Here, only a representative patient is plotted for illustration purposes.

The choice of orienting the robot vertically alongside the patient, was made after a workspace analysis of the *Novint Falcon*^{TM}. It was visible that the major axes of free displacement are the Ox and Oy and, as such, it matches the biggest displacements needed. According to the analysis of data, a displacement of around 15mm in Ox and Oy was needed in order to perform all the registered biopsies, while in Oz only a 6 to 7mm displacement was obtained.

4.3 Algorithm

There are many different optimization algorithms in the literature to find a maximum of a constrained non-linear function. The goal is to maximize the manipulability in the points obtained from the data acquisition. For this, Equation 4.2 is the cost function to maximize, applied to the inverse kinematic model of each one of the cloud points. However, after trying some algorithms such as the Gradient-Based Method implemented in Matlab by the *fmincon* function [28], it was evident that it was very difficult for Matlab to process this gradient.

Algorithm 1: Discrete Iterative Search for Greatest Average Manipulability

```
Input  : Biopsy points cloud
Output: Parameters for the greatest Average Manipulability measure
For each a = minimum a : step size : maximum a , do
 For each b = minimum b : step size : maximum b, , do
  For each d = minimum d : step size : maximum d, , do
   For each r = minimum r : step size : maximum r, , do
    For each pv0 = minimum pv0 : step size : maximum pv0, , do
     For each z = minimum z : step size : maximum z, , do
      Assign robot parameters $= [a,\ b,\ d,\ r,\ pv0,\ z]$;
      For each p = first biopsy point : last biopsy point, , do
       $p_z$ = $p_z$ + parameter $z$;
       Compute Inverse Geometric Model $\theta = L^{-1}(\mathbf{p})$;
       If Biopsy point is within workspace , then
        Increment counter of points reached by these parameters' robot;
        Compute Manipulability measure $w(q)$ of this point for these parameters;
       EndIf
      EndFor
      Compute Average Manipulability measure of the set of points;
      If Average Manipulability measure is greater than previous , then
       Save greatest Average Manipulability measure;
       Save mesh point parameters;
      EndIf
     EndFor
    EndFor
   EndFor
  EndFor
 EndFor
EndFor
```

One can imagine the extent of the final expression of the manipulability measure, its differentiation symbolically was very difficult with this implementation. Not only this, but this difficulty was multiplied by applying this expression to 10954 biopsy points, considered as constraints for the gradient. Also, the large number of parameters to optimize may have had a negative effect on this computation. This because the gradient may have not been capable of isolating the parameter in the cost function.

A simpler method was chosen to realize this optimization, which provided a better control of the physical meaning of the optimization parameters. This approach was a discrete iterative search with a double refinement in the mesh values to check for convergence.

Another difference in the method adopted is that the cost function is computed for all the biopsy points in one parameters' mesh value. Only after is the true cost function derived by the average of the manipulability measure for all points in this mesh value, that are within the workspace of the robot. This procedure is used to avoid local maxima in finding the maximum manipulability for all mesh values. In reality, a maximum average manipulability is optimized, and the number of points reached is accounted for. The algorithm implemented is described in Algorithm 1.

4.4 Results

In this section, the results of the optimization described in Algorithm 1 are presented. The first run of the algorithm was an iteration where the set of values that each parameter could take are presented in Table 4.2. The results obtained for this first run, where a rough mesh was used, are represented in Figure 4.3. This picture represents the obtained set of parameters which maximize the average manipulability measure, for each value taken by the parameter a.

Parameter	Min Value	Max Value	Step Size	Nº Steps
a [mm]	60	150	10	10
b[mm]	100	300	20	11
d=e [mm]	10	30	10	3
r [mm]	50	150	10	11
||pv0|| [mm]	20	100	10	9
z [mm]	100	600	100	6

Table 4.2: First possible values of the optimization parameters for the first run of the iterative search.

Analysing the results, one can see that there were two mesh points where the average manipulability reached maximums. Those points correspond to the parameter sets defined by $a = 110,\ b = 140,\ d = 20,\ r = 140,\ pv0 = 60, z = 10$, and $a = 140,\ b = 140,\ d = 30,\ r = 90,\ pv0 = 50,\ z = 10$, in milimeters. The manipulability measure is high for both sets, but while the first set reached only 10850 biopsy points out of 10954, the second set reached 10938, the highest number of points representing 99.85% of all the points cloud.

For the reasons presented, the second set of points will be considered the optimal solution for this mesh. However, another iteration must be done with a more refined range of parameters in order to prove convergence of the algorithm.

The second run of the algorithm represents a refinement of the mesh of parameter values. The range of values that each parameter could take are again redefined and presented in Table 4.3. The results of the optimization for the refinement are plotted in the graph of Figure 4.4.

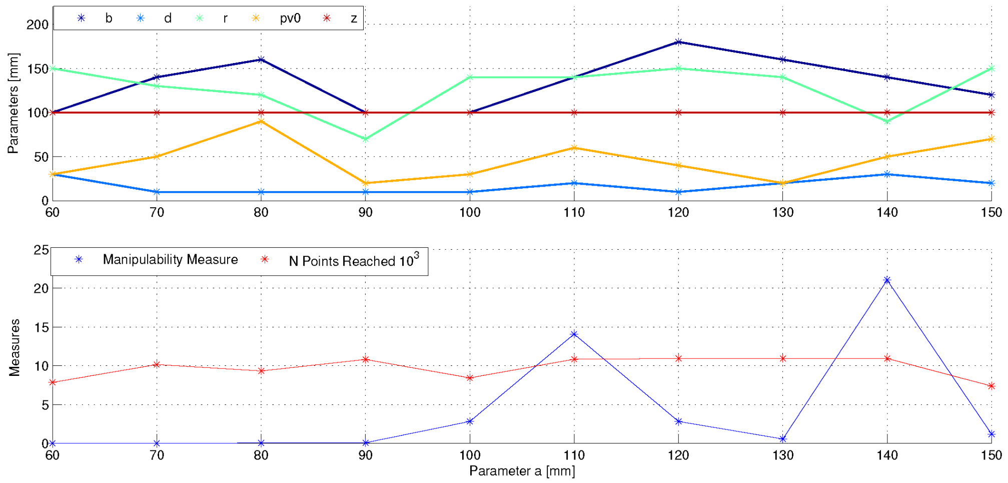


Figure 4.3: Graph representing the results obtained for the first run of the iterative search. Top plot: Optimization parameters b, d, r, pv0 and z, displayed in terms of the evolution of parameter a, in the x axis. Bottom plot: Maximum average manipulability measure and points reached obtained for each set.

Parameter	Min Value	Max Value	Step Size	Nº Steps
a [mm]	135	145	5	3
b[mm]	130	150	10	3
d=e [mm]	25	40	5	4
r [mm]	85	95	5	3
||pv0|| [mm]	45	55	5	3
z [mm]	100	200	100	2

Table 4.3: Possible values of the optimization parameters for the refinement run of the iterative search.

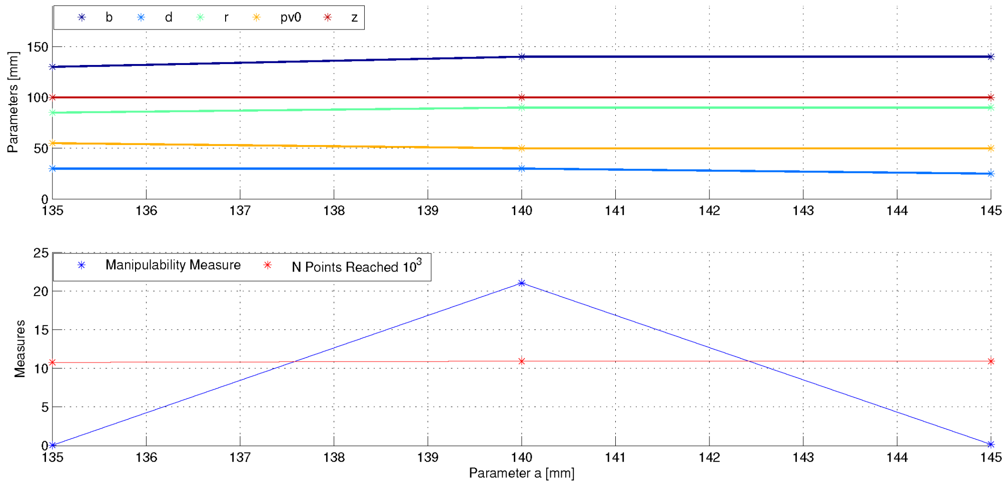
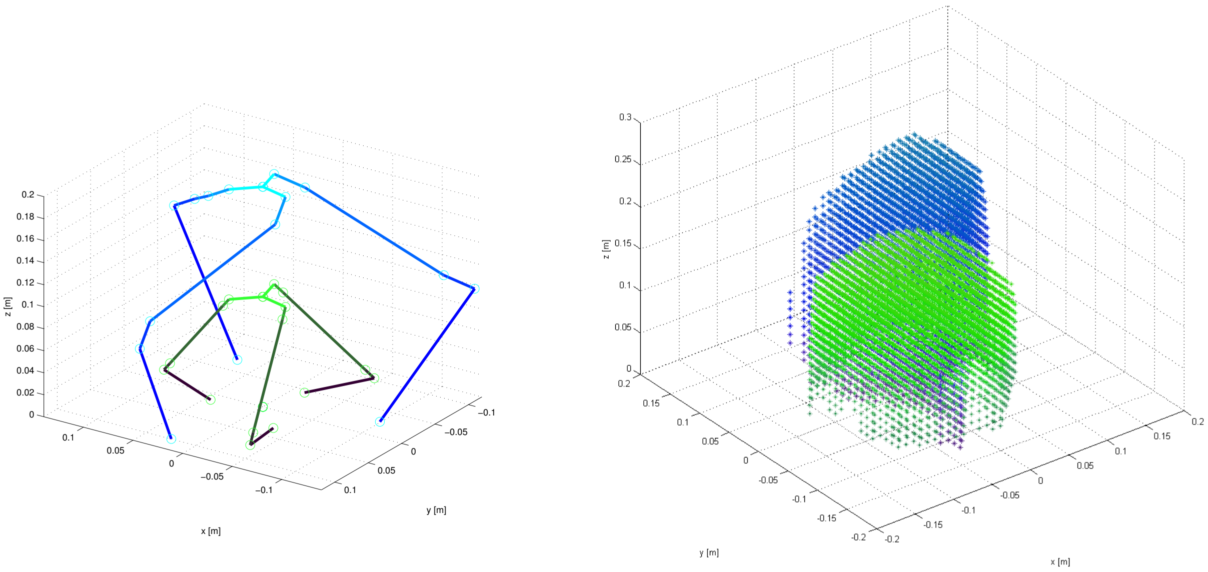


Figure 4.4: Graph representing the results obtained for the second iterative search. Top plot: Optimization parameters b, d, r, pv0 and z, displayed in terms of the evolution of parameter a, in the x axis. Bottom plot: Maximum average manipulability measure and points reached obtained for each set.

After the analysis of the refinement results around the first original optimal point, the convergence of the search is proved. The range of values that the parameters could take was shortened and the step size reduced to half. The maximum point of manipulability obtained was the same as in the first iteration, corresponding to $a = 140,\ b = 140,\ d = 30,\ r = 90,\ pv0 = 50,\ z = 10,$ in milimeters, with equal values of average manipulaility measure and points reached.

One can argue that a deeper refinement would be needed. However, because these parameters are to be applied in the design of the new structure, the tolerance of around 5mm in each parameter can be considered acceptable. This is because the methods of manufacture adopted to build the robot are not so precise as one would wish, and this tolerance is relatively small compared to the new size of the links.

The relative proportions of the original robot and the optimized one can be compared in Figure 4.5(a), and the relative size of the workspace of both structures can be seen in Figure 4.5(b).



(a)

(b)

Figure 4.5: (a) Representation of the optimized structure (in blue) at the center of its workspace, comparatively with the original one (in green). (b) Plot of the workspace of the optimized structure (in blue) and the original one (in green), without considering joint limits and collisions.

Analysing Figure 4.5(b), it is possible to compare the cartesian ranges of the workspace of both kinematic structures. In green, the original workspace of the Falcon Novint, presents a workspace range in the x and y axes of $[-0.1; 0.1]$m in a total of 20cm and in the z axis of $[0; 0.15]$m in a total of 15cm. In blue, the optimized workspace of the new robot shows a wider range in all the axes: from $[-0.15; 0.15]$m in x and y, in a total of 30cm, and in the z axis of $[0; 0.25]$m, in a total of 25cm.

However, it is necessary to take in consideration that the format of the workspace is oval-shaped, due to the singularities of the three arms. Taking this in account, it is visible that the workspace was stretched more in the positive z direction and fewer in x and y. This is coherent with the data since during a biopsy, it is more necessary to move the probe in order to rotate the ultra-sound sensor in the sagittal plane of the patient, than to displace the sensor.

5 | Implementation

After obtaining the results for the kinematic optimization in Chapter 4 it is necessary to plan the design and manufacture of the robot as well as to implement its communication with the desired software and incorporate the necessary sensors so that the robot may be controlled.

5.1 Design

Starting with the design of the robot, the final result of the design of each part and the methods used to fabricate it, will be presented. When relevant, design decisions will be adressed.

5.1.1 Links

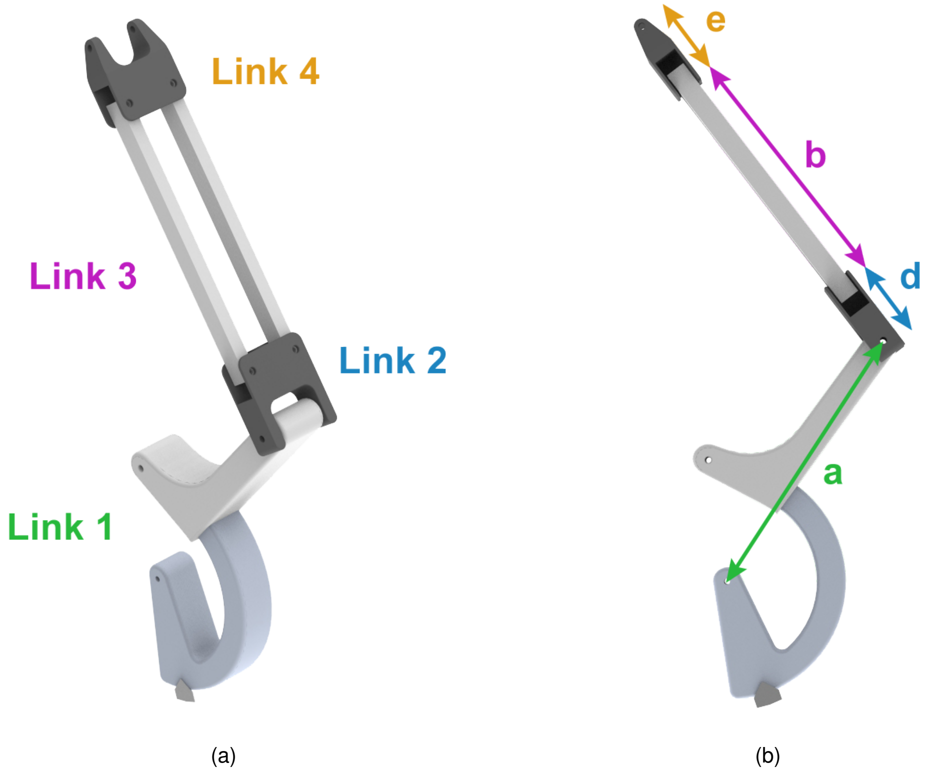
The links consist of the rigid parts connecting each joint, in this case, revolute joints. After the kinematic optimization, the robot links need to be designed according to the obtained measures. Following the same principle as the *Novint Falcon*, the generic arm consists of: a circular link (Link 1); an adapter from the first joint to the parallelogram structure (Link 2); the two parallelogram bars (Link 3); the adapter from the parallelogram to the end effector (Link 4) and the end effector itself, as represented in Figure 5.1(a).

The original circular link from the *Novint Falcon* was used for the first link because it already provided a good transmission system to the motors through cables, and contained a system with a spring to tension the cables. However, because the dimension of the radius (a) of the first joint obtained through optimization was bigger than the original, a radius adapter was designed. This adapter consists of two parts that assemble together around the tip of the original link as represented in the CAD display on Figure 5.1(a).

The rest of the links are relatively simple and similar to the original *Novint Falcon* ones, and thus won't be detailed but their dimensions can be seen in Figures 5.1(a) and 5.1(b).

5.1.2 End Effector

The top part of the robot, where the three arms connect is referred to as the end effector. The original end effector of the *Novint Falcon* was a grip ball with multiple buttons, however these were unnecessary for the current robot and were thus discarded. The only part which was recycled was the base where the joint connections are assembled.



(a)

(b)

Figure 5.1: (a)Isometric view of one of the robot's arms and specific links. (b) Side view of one arm with geometric parameters.

For the prostate biopsies, as referred in Chapter 1.2, a TRUS probe is used. To add the manipulation of this probe to the 3 translational DOFs of the robot, it was necessary to create an interface for the end effector. This interface for the TRUS probe consists of a spheric passive joint, to be actuated by the urologist according to the necessary orientation for the probe, inside the patient. This adds 2 rotational DOFs to the robot, now with a total of 5DOFs.

The implementation of this spheric joint was made by 3D printing a vertical arch connected to the base of the end effector by a 608ZZ single row bearing, and two threaded rods connected to the each side of the arch by a 624ZZ single row bearing. These rods were then screwed to the part holding the probe itself and fixated accordingly on the arch so that the position of the probe was centered. The total assembly of the end effector can be seen in Figure 5.2.

5.1.3 Robot Encasing

In order to follow the parameters obtained by the output of the kinematic optimization in Chapter 4, it is needed to assemble the robot with some geometrical precision. The most important parameter defined by the assembly of the three arms will be the **r** parameter, which approximately represents the radius from the center of the base to the first joint of each arm.

Thus, a solution to maintain these relationships without the need for high calibration systems in perforation and fixation, which are usually very expensive and time-consuming, was needed. The result was the use of the inexpensive laser cutting method, which enabled the fabrication of a support box

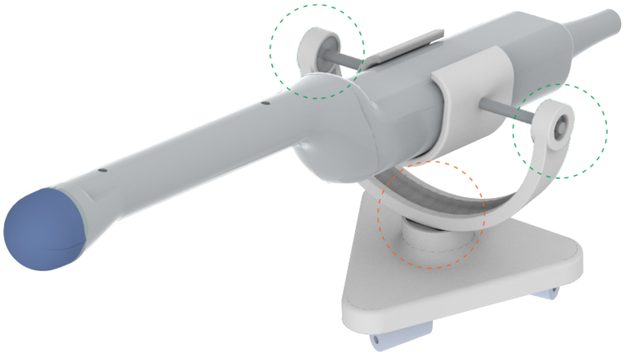
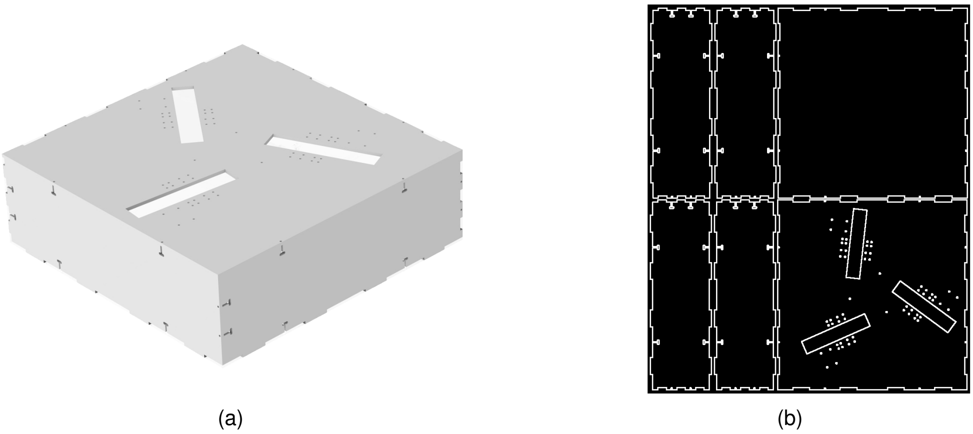


Figure 5.2: View of the spheric end effector model with TRUS probe attached. The circles indicate the placement of the bearings.

for the robot. This encasing not only contains the desired holes to fixate the arms with respect to the geometric constraints, but also the motors and sensors and an exterior protection for the electronics.

The laser cutting layout and the model of the referred encasing can be seen in Figure 5.3(b) and 5.3(a), respectively. The walls are to be bolted with a type of jointing similar to those of today's generic 3D printers.



(a)

(b)

Figure 5.3: (a) Robot encasing 3D model. (b) Laser cutter layout of the parts.

5.2 Actuators

The actuators used for the robot were recycled from the *Novint Falcon* and can be seen in the original setup in Figure 5.4. There was a great lack of information about the Falcon's specifications since the Novint Technologies, Inc. corporation became bankrupt and all of the data repositories are offline. However, from data obtained from online reports [29], it was possible to know that the *Novint Falcon* was powered by a 30V, 1A transformer. From this information, one can extrapolate that the maximum voltage to be given to the DC motors was 30 V.

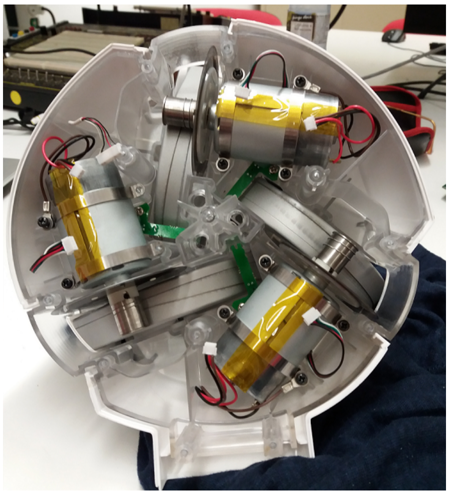


Figure 5.4: Original setup of the three actuators with optic encoders for home position detection.

Each of the three motors had a ground connection that was left unconnected, and their speed control was achieved by pulse-width modulation (PWM) through the positive and negative inputs.

The final assembly with all the previous referred components, such as the arms, end effector and encasing can be seen modelled in Figure 5.5. The built prototype with the corresponding links, acrylic fixations, sensors, actuators and micro-controller can also be seen in Figure 5.5, right side.

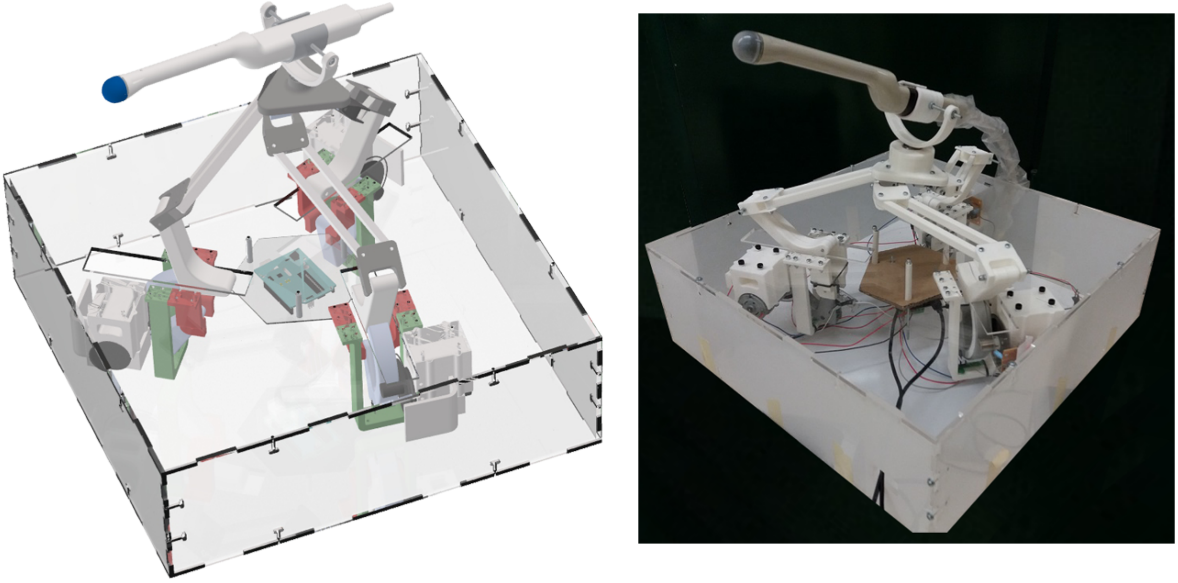


Figure 5.5: CAD model and prototype of the final assembly with encasing, arms and end effector.

5.3 Arduino Board

When the mechanical implementation was already detailed, it was time to consider the electronics. The first approach was to try and re-use the original *Novint Falcon*'s electronic board. In order to this, it was necessary to access the Software Developer Kit (SDK) provided by *Novint* in order to change the kinematic parameters coded into the original device, and also, to understand the boot system and homing sequences that were used to set all the sensors and actuators.

However, after a lot of time spent with these studies, it was concluded that some intrinsic software parameters could not be modified because some classes were only available in compiled, black-box versions. Thus, it was impossible to understand how the Novint system would boot, and how it was possible to change the pre-defined communication. This is because the original haptik device had as inputs to the micro-controller, cartesian forces to be applied at the end effector, which meant that at least the kinematic model was already pre-implemented. In our application, it was desirable to have as inputs the position or velocities to be applied to the motors.

In order to achieve this, the original board was discarded and an *Arduino Uno*^{TM} was selected. The *Arduino Uno*^{TM} is a micro-controller based on the ATmega328P, and it has become a very popular item for prototyping electronic projects due to its simplicity, C++ based interface and low cost. It has 14 digital input/output (I/O) pins, 6 with PWM outputs, a 16MHz quartz crystal, USB connection for serial communication and power supply, and a power jack in case there is the need for external power [30].

However, the technical specifications for the *Arduino Uno*^{TM} advise a maximum input voltage of 12V, with a 5V operating voltage, and a maximum DC current per I/O pin of 20mA. As referred in Section 5.2, the DC motors used have a nominal operating voltage of 30V, and 1A current. For this reason, there was the need to use an adaptor for the Arduino that allowed a greater power supply. The solution was the *DFRobot*^{TM} *2x2A DC Motor Shield*.

This shield is mounted on top of the Arduino, its pins are connected to the Arduino pins, and its operating voltage of 5V is also provided by the Arduino. It uses the L298P chip which allows the shield to drive of up two DC motors with up to 2A per channel. Also, using an external power supply, the Arduino pins 5 and 6 are used for PWM speed control, and the pins 4 and 7 for direction control of the two motors [31].

Because each shield only allowed the control of two motors, two shields were implemented on top of the Arduino board to control the robot's three motors. In order to avoid conflicts between the use of the same pins for the two shields, an intermediary board was used to connect the PWM and direction control pins of the second shield, to the digital pins 2 and 3 of the Arduino. A summary of which pins are connected is shown in Table 5.1.

Apart from allowing to send information to the motors, the arduino board also allows to receive information through digital pins from the encoders used in this robot, as will be detailed in the next section.

5.4 Sensors

This section will discuss the implementation of sensors used to detect the movement of the motors, as well as to count the rotations done by them. This is very important, because if one can not measure the angular displacement of the robot and the movement transmission ratio, then it won't be possible to know how much the angular joints displaced and successively, the end effector.

Arduino Pin	Function
Digital 4	Motor 1 direction control output.
Digital 5	Motor 1 PWM control output.
Digital 7	Motor 2 direction control output.
Digital 6	Motor 2 PWM control output.
Digital 2	Motor 3 direction control output.
Digital 3	Motor 3 PWM control output.
Digital 8	Encoder 1 signal input.
Digital 10	Encoder 2 signal input.
Digital 12	Encoder 3 signal input.

Table 5.1: Connection between the Arduino's digital pins and the Motors' PWM and direction outputs, and also sensor inputs.

In this approach, only two optical encoders were assembled for each arm. One encoder was an incremental one, with an encoder disk, that enabled the count of the number of angular displacements defined by each breach in the disk, as seen in Figure 5.6(b). The second encoder was to be used as a homing position sensor when a plastic tip would close the circuit. However, the later one was assembled but not implemented and can be used in future works, represented as the green support in Figure 5.6(a).

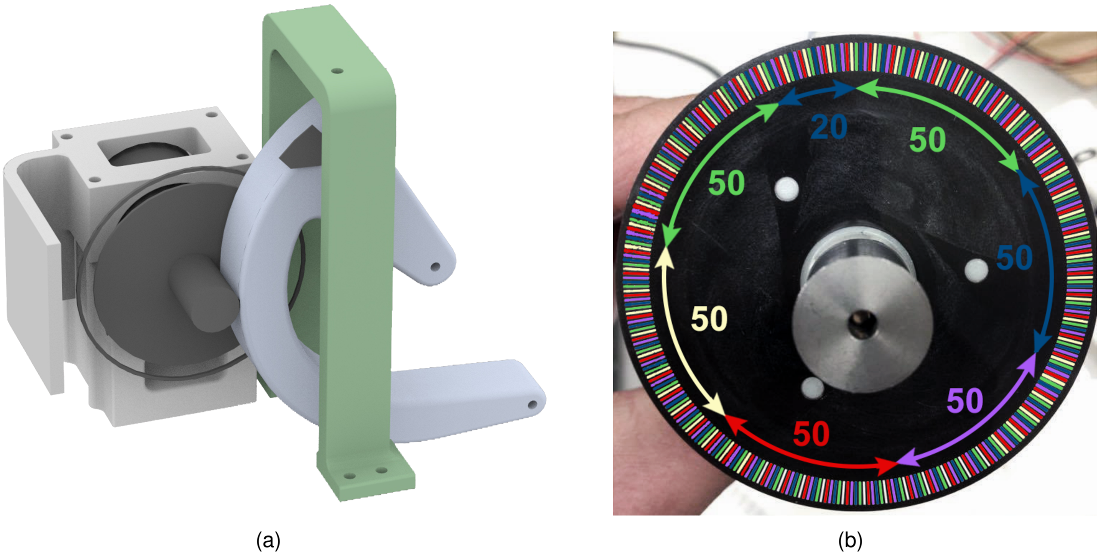


Figure 5.6: (a) 3D model of the placement of both encoders. (b) Number of divisions per encoder disk = 320.

The original Falcon included a light source emitter and a receptor for each motor encoder disk, however, their re-utilization was not possible. Although the reasons are unknown, the original sensors didn't emit any signal after being de-soldered from the main board. Thus, a new photointerrupter sensor was acquired for each one of the three disks (GP1S53VJ000F by *SHARP*). The method of selection was the lowest cost possible for the bigger precision defined by the emitter slit size, that was obtained as 0.5mm.

The measured signal for each encoder setup was a sinusoidal wave with the frequency proportional to the angular velocity of the motor, with peaks according to the number of breaches that the photointerrupter registered. However, the maximum amplitude of the wave was around 4.8V while the minimum was 3.3V, for high rotational velocities. This means that the peak to peak amplitude was around 1.5V, as can be seen in the yellow signal of channel 1 in Figure 5.9.

As referred in Section 5.3, the micro-controller used was an Arduino Uno board. This board reads digital input signals as HIGH or LOW if the reference voltage of the signal is higher than 3.0V or lower than 1.5V, respectively [32]. This meant that the actual sinusoidal signal coming from the raw circuit of the encoder didn't suffice for the Arduino to read the number of angular displacements correctly.

5.4.1 Amplifier Circuit

For the aforementioned reason, it was necessary to develop an amplifying circuit that would increase the peak to peak voltage of the signal, while maintaining a certain low voltage level for the LOW threshold. After some analysis of the signal and of the components available at the laboratory, without the need to spend much, the Encoder circuit shown in the upper-right corner of Figure 5.8 was designed.

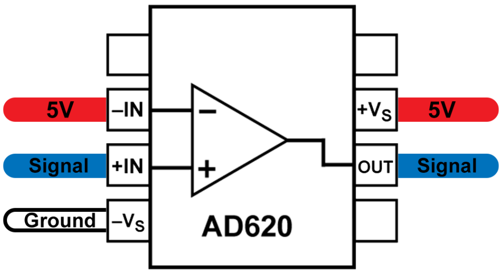


Figure 5.7: Inputs and Outputs for the AD620 high accuracy instrumentation operational amplifier used in the encoder circuits.

In the lower left corner, it is visible that there is a power supply (connected to an electrical socket) that provides power to the *DFRobot^{TM} 2x2A DC Motor Shield*. This shield then provides power to one or two motors, depending if it is the first shield, or second, respectively. As referred, each motor has an optical encoder sensor which is also connected to an amplifying circuit, powered by the 5V output pin of the Arduino, and that outputs the encoder signal to the arduino input pins 8, 10 and 12, respectively.

The amplifying circuit has the purpose of transforming the raw signal from the encoder. The objective was to reduce the minimum voltage offset, and amplify the peak to peak voltage into approximately 5V. After some testing, the final result was the use of the high accuracy instrumentation operational amplifier (opAmp) AD620 with the inputs as represented in Figure 5.7.

The idea was to use the 5V voltage as the negative input to be subtracted to the raw signal from the encoder. However, this specific opAmp is rather complex and already has incorporated micro-resistances and gains. Thus, this final setup was obtained experimentally by a trial and error sort of approach, while remaining faithful to opAmp theory.

With this amplifying circuit, it was possible to increase the peak to peak average voltage to approximately 3.3V, with a maximum of 4.3V and a minimum of 1V as can be seen in the blue signal of channel 2 in Figure 5.9.

These values were thus much more suitable to be read by the Arduino digital input pins, even when the frequency of the motors was highest. However, there are still some oscillations in these values from motor to motor. There are several reasons for this: one is that not all components are exactly equal and some were recycled from what was available, being subject to some use during previous years that could have degraded their functionality; Also, the approach used to fixate the circuits to each encoder disk was very challenging, due to the fact that each disk has its optimal sensor position, thus making each fixation system unique; Another reason for peak to peak fluctuations is the relatively bad condition of the encoder disks, which was already present in the Falcon assembly, where several bulging can be visible in their thickness, causing the phenomena registered for a low frequency update rate in Figure 5.10. Even after all these considerations, the Arduino board was able to register all encoder rotations, which was verified even for higher rotations of the motors.

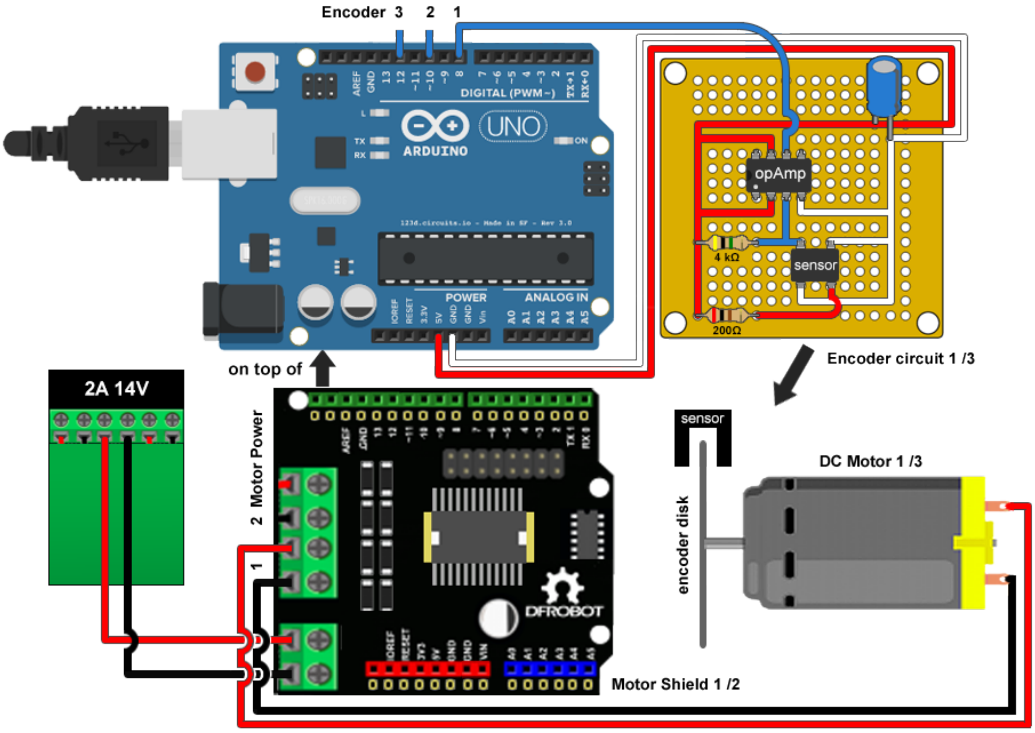


Figure 5.8: Diagram of the electronic assembly between the Arduino board, DFRobot shields, optical enconders, motors and the amplifying circuits.

5.5 Serial Communication

Lastly, it is necessary to approach the methodology used to send and receive information from the Arduino board to the Computer, which in this case was implemented in MathWorks' Simulink®. Simulink is a block diagram environment for simulation, control and communication project design.

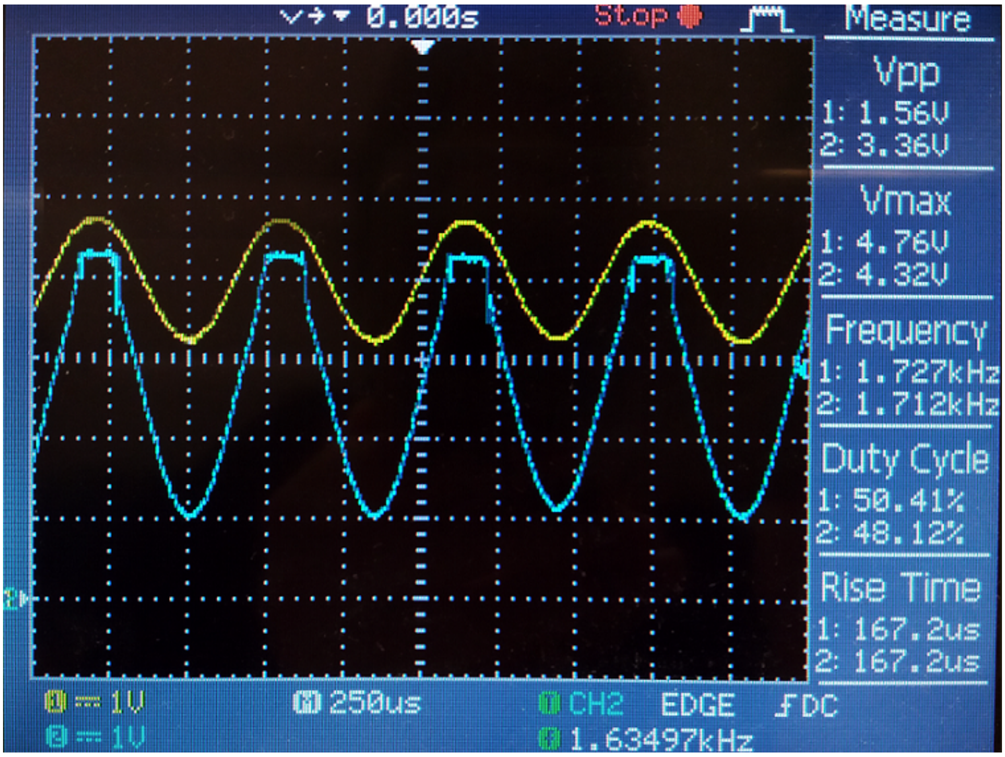


Figure 5.9: Oscilloscope registration of the raw signal coming from the optical encoder (channel 1 in yellow) and the amplified signal, output from the circuit (channel 2 in blue).

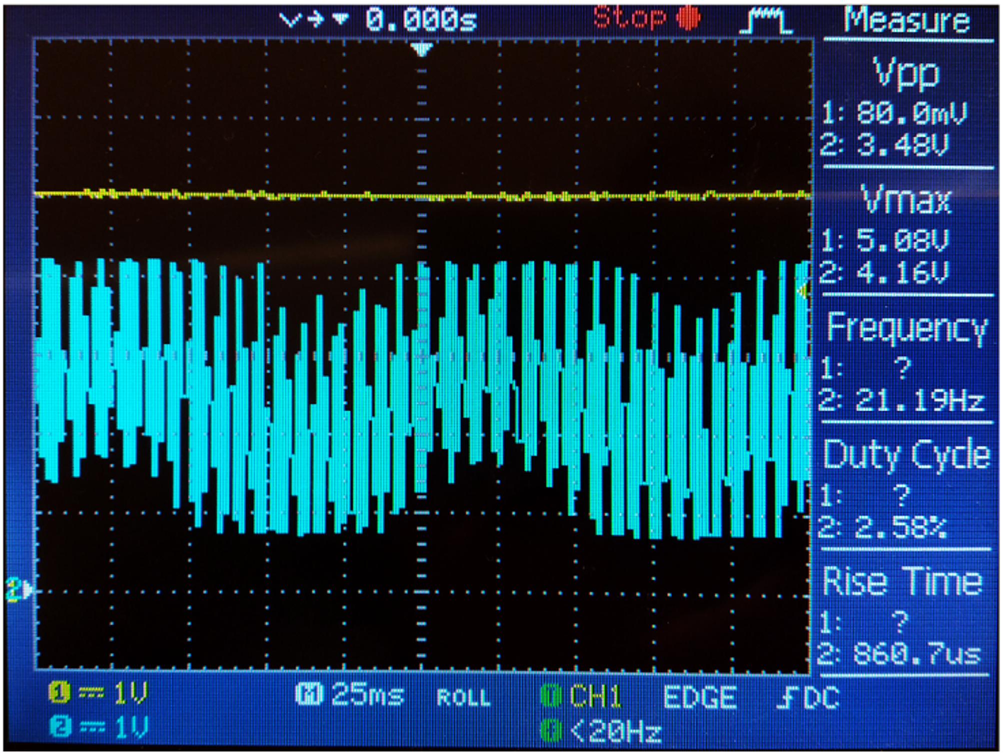


Figure 5.10: Oscilloscope registration, with 25ms update periods, where the bulging of the encoder disks can be seen to affect the long scope wave (channel 2 in blue).

The protocol used for communication was serial, which sends byte by byte information, corresponding to 256 bits each. In this project it was necessary to send to the Arduino the desired velocities to each one of the three motors, which was a number between 0 and 255, and the desired direction, which could be 1 or 0 to rotate clockwise, and anti-clockwise, respectively.

This represented a total of 6 numbers to send, which were sent in 6 different bytes. This approach could be optimized in order to send all the directions in one single byte, given that for each, only one bit is needed. However, this was not implemented due to the difficult and time consuming debugging of the serial communication. A header byte to flag the start of the message was defined as the decimal number '17', randomly chosen as not likely to be repeated.

Also, it was necessary to receive from the Arduino board, the readings of each one of the encoders, so as to count the rotations of the motors. For this, a routine named interrupt was used for each of the input Arduino pins.

The routine used was available from the 'EnableInterrupt' open source Arduino library, which provided classes to implement triggering of pin change events such as RISING or FALLING signal edges on digital input pins [33]. When an event triggers an interrupt on any interrupt-enabled pin, a subroutine (ISR) attached to that pin's port is triggered. In this case, whenever an encoder registers one passing of a disk's slit, a RISING edge is detected and the ISR checks the direction of the motor, and increments or decrements the rotation counter, accordingly.

For the Arduino board communication it was defined a period of 1000 micro-seconds, or 1 millisecond to serial send the counters defined by 16bits integers as two bytes, each. The communication block designed in Simulink can be seen in Figure 5.11.

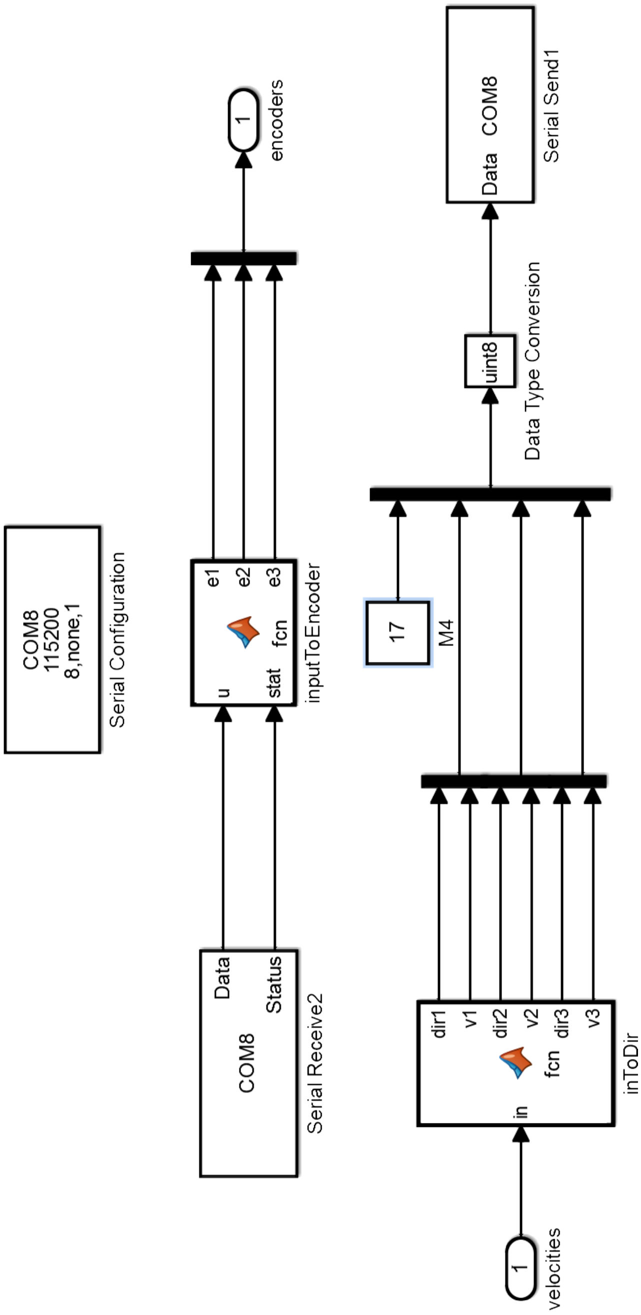


Figure 5.11: Simulink® diagram block for serial communication. Above, the inputs from the Arduino, then translated to encoder rotations. Below, the ouputs from the model as velocities to the motors.

6 | Control Planning

In this chapter, the approach used for the control design is presented. After planning the modelling, optimization, and construction, it was time to prepare the control solution to be implemented.

6.1 Control Model Study

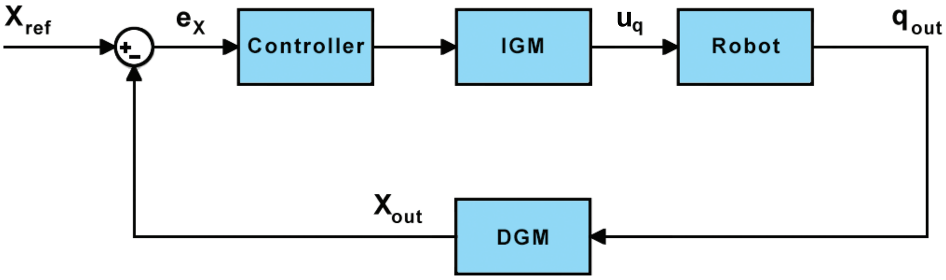


Figure 6.1: Diagram of cartesian position control model of a generic robot.

In Figure 6.1 a generic diagram of a position control model is presented. Desired values of end effector cartesian coordinates are provided as reference (X_{ref}), and the actual cartesian coordinates read from the robot are evaluated through feedback. After passing the controller, the input is transformed through the Inverse Geometric Model (IGM) to the joint space of the robot and transmitted to the actuators. The feedback of the actual joint coordinates is also transformed to the cartesian space through the Direct Geometric Model (DGM), in order to compare with the reference position.

At this point in the project, the idea was still to use the Novint Falcon's original board and software. In order to test that it was possible to establish a control loop with this, initial tests were done on the original Falcon using an open source communication library that enabled the link between the Falcon and Simulink, called the *Haptik library* [34].

Haptik is a small component-based library to access haptic devices in an uniform way. It implements the Falcon's SDK (Sofware Developper Kit) in its plug-in. However, the inputs to the robot are cartesian forces of the end-effector (since it was originally a haptik device) and the output is its cartesian position. This means that the Inverse Dynamic Model (IDM) of the original structure is hard coded as well as its DGM. A schematic of what the *Haptik library* plugin does is represented in Figure 6.2.

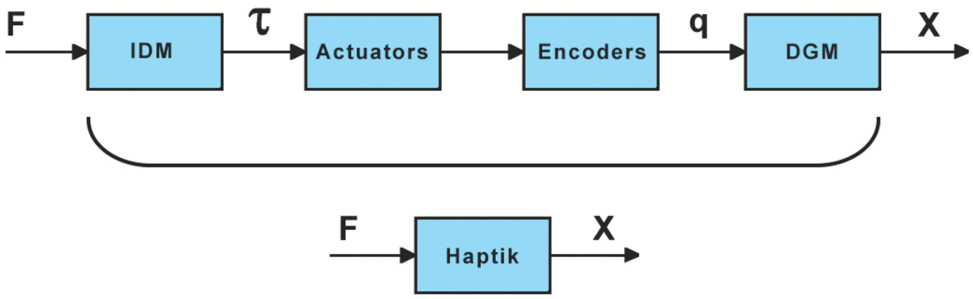


Figure 6.2: Diagram of the *Haptik Library* Falcon plug-in.

An initial feasibility control test was prepared with the original haptik device, which implemented a simple feedback loop and a proportional controller. The theory behind this approach was that in order to control the cartesian position of the robot, since we already had access to cartesian force inputs, a proportional relationship existed and could be controlled through a simple gain.

This was such that the further away the robot needed to move, or the bigger the difference between desired and actual coordinates was, a proportional value was transmitted as what force needed to be actuated in each direction. To better understand, the control implemented is represented in Figure 6.3.

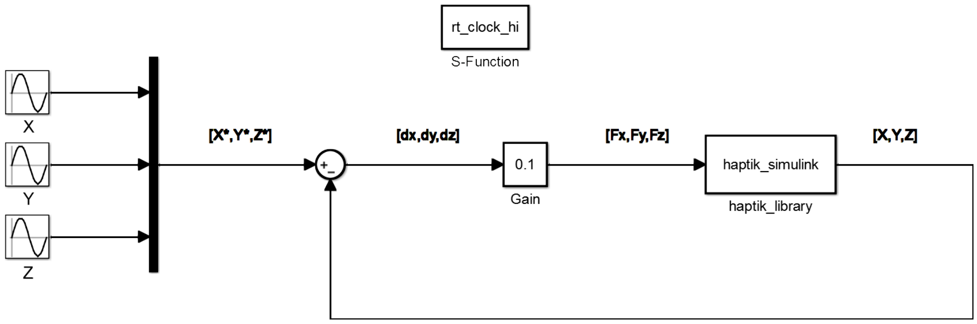


Figure 6.3: Simulink block diagram of the control model implemented with the Haptik Library for the position control of the original Falcon.

The inputs to the X^*, Y^* and Z^* desired positions could be determined previously and uploaded to the block diagram. The gain applied was determined experimentally by tuning of the P controller. This was mainly due to the complexity and black-box properties of the SDK inherent model.

The ’rt_clock_hi’ block enabled the synchronization between the time step of the Simulink solver and the real time, by means of a solver delay. If this solution was satisfactory, one could extrapolate the same feasibility to the actual control with the Arduino board.

6.2 Results

The results obtained from the control model explained previously will be presented in this section. As referred, the objective is to obtain a proof of concept of the control model with the original Falcon.

The criteria used to choose the trajectories was to have a good mechanical stability of the manipulator, while not applying sudden high forces. Because of the limited workspace of this kind of kinematic structures, the trajectory chosen was a circular one.

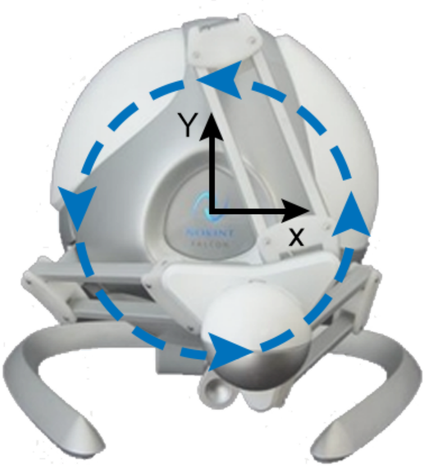


Figure 6.4: Representation of the desired trajectory in the xOy plane as a circular/ellipsoid movement.

The movement was defined as an ellipse in the xOy plane as represented by the desire movement in Figure 6.4. First, a cosine wave was given as input to the X desired position, a sine wave to the Y position, and an almost null sine wave was defined for Z. these results are shown in Figure 6.5.

After, just as a matter of robustness verification of the control law and results, the inputs where switched so that the desired position for X became a cosine and Y position a sine. The results for this last test are shown in Figure 6.6.

More specifically, the sinusoidal waves used as inputs to the control model were generated such that the device handle would remain in the workspace of the haptik device. Because the original workspace had an amplitude of around 60mm to each direction of x and y, a desired amplitude of 40mm was defined in this direction, while in z the amplitude was almost null. Also, the frequency was chosen such that the movement was observable and smooth relatively to the mechanical constraints of the structure. A velocity of around 0.5 rad/s was chosen which is roughly equivalent to 30 deg/s.

After analysing the results, one can say that on both cases the system followed reasonably well the desired position. However, there is still room for improvement. For example, there is clearly a big delay between the actual position and the desired input, which could be improved, for instance, with the use of a derivative term in the controller. Also, the proportional term needs to be more finely tuned to adjust the magnitude of the output, and an integral term could be used to eliminate any steady state errors that may not be noticeable in the current output.

6.3 Proposed Control Model

In this section, the control model to be used in the optimized and newly-built robot is to be discussed. The feasibility test, explained in the previous section, was done under the assumption that the original communication system was to be kept as well as the micro-controller board. This was in the beginning of this project, however, for the reasons mentioned in Chapter 5.3, this electronic board had to be replaced

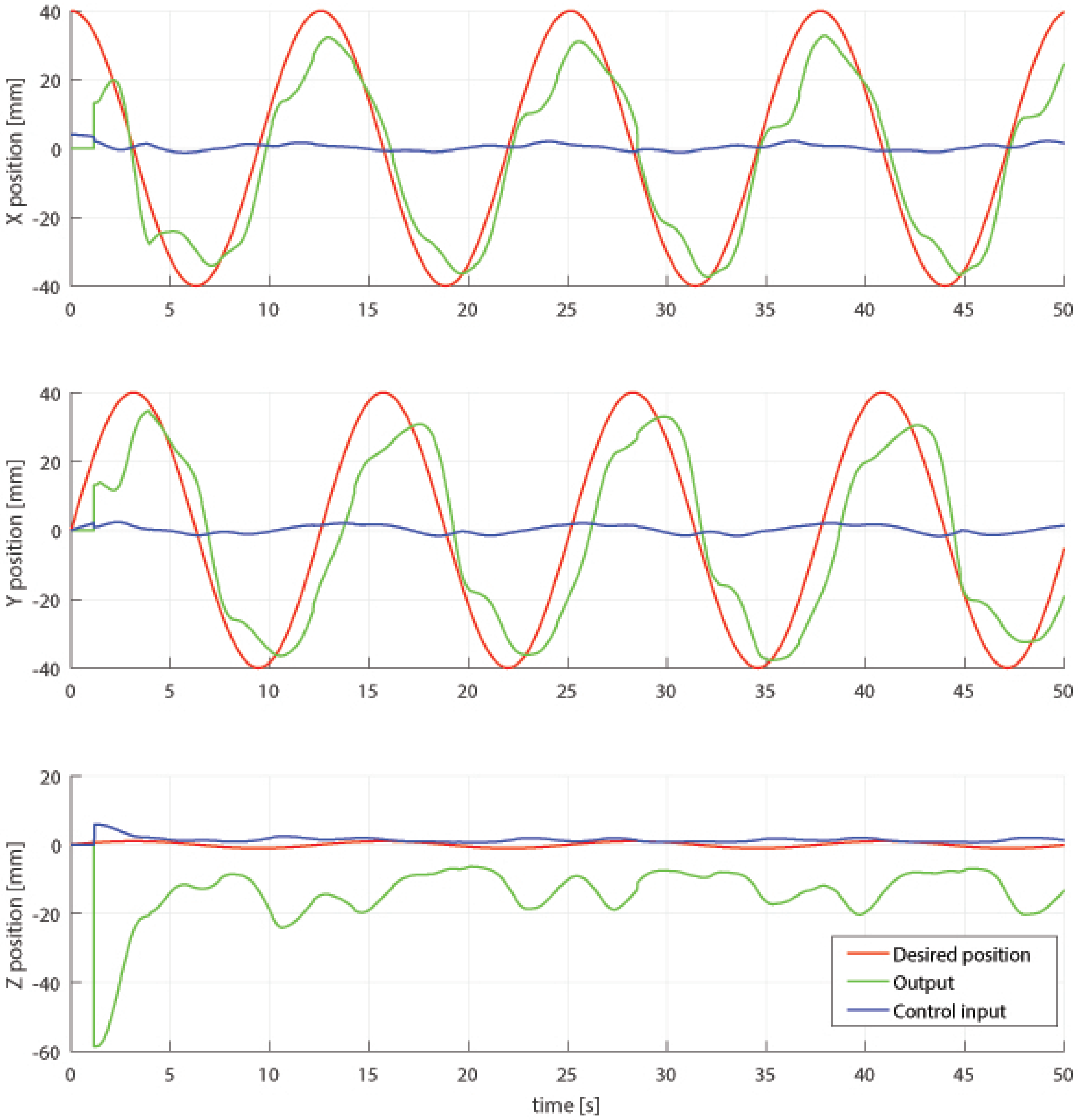


Figure 6.5: Results obtained for the control study with the follow desired positions: $X^* = 40\cos(0.5t)$, $Y^* = 40\sin(0.5t)$, $Z^* = 0.05\sin(0.5t)$.

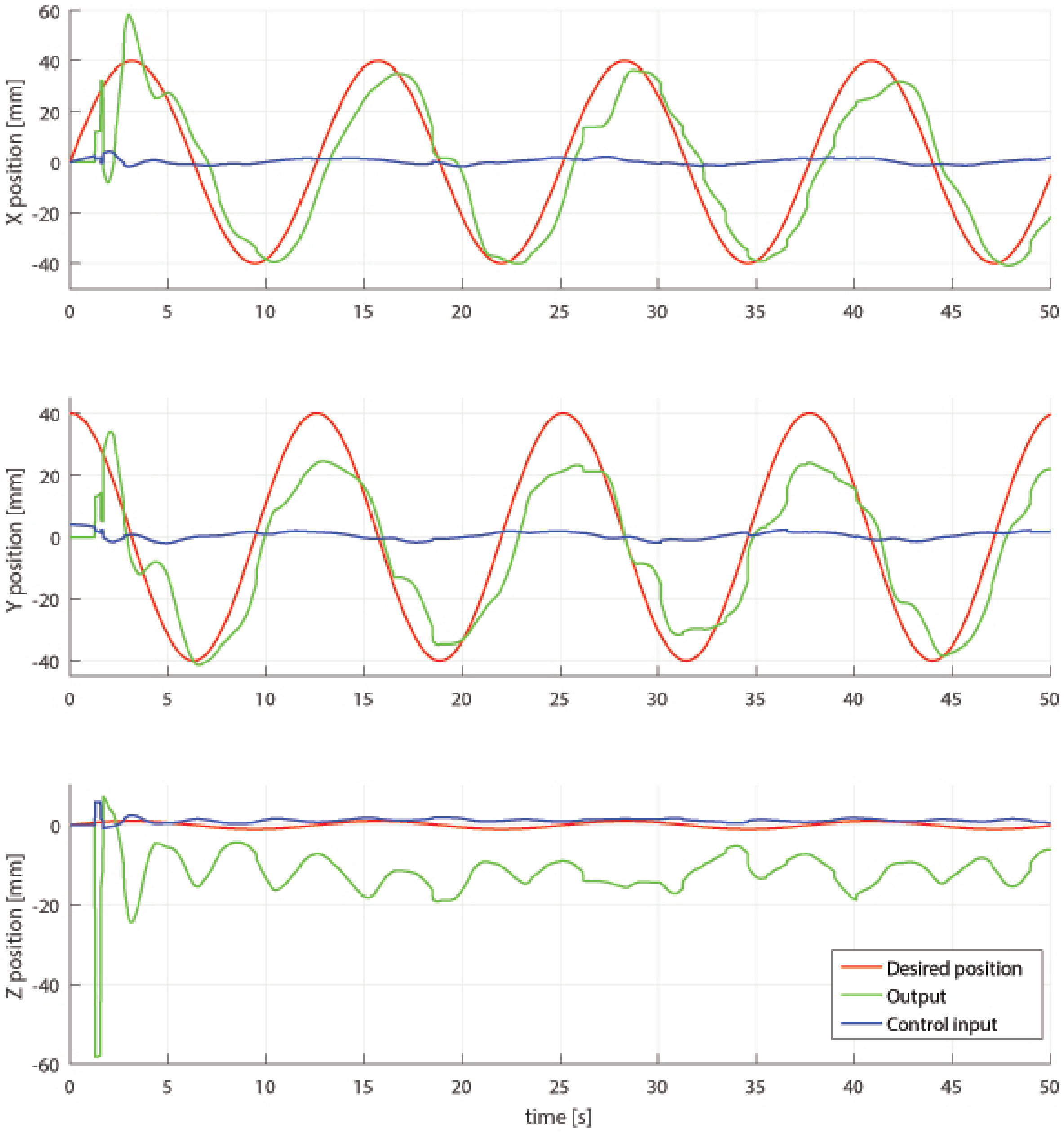


Figure 6.6: Results obtained for the control study with the follow desired positions: $X^* = 40\sin(0.5t)$, $Y^* = 40\cos(0.5t)$, $Z^* = 0.05\sin(0.5t)$.

by an Arduino board.

The original board contained, embed in the micro-controller code, the voltage control for the applied force of each one of the motors. This means that there was some kind of dynamic model, and inner speed controllers, probably PID, for the motors. After removing the original board, this voltage control for the speed of the DC motors was lost, as well as the hard coded kinematic and dynamic models. This meant that to control the final, optimized and rebuilt robot, a more complex design is needed than the one used on the feasibility test.

In this project, the purpose was to control this kinematic structure in order to obey the requirements computed from an MRI and TRUS-based image fusion, as proposed in Chapter 1.5. In the end, this means to send desired positions and trajectories to the end effector in order to constrain the movement of the urologist in the direction of the prostate. To do this, there isn't the need for the design of torque or force control. Having this in mind, the proposed control solution will be a normal position control in cartesian space, transforming to the joint space through the IGM, with specific control of the rotational speed of the motors in a inner loop.

The methodology proposed to control each one of the DC motors is presented in Figure 6.7, in the blue sub-diagram. The input is a desired velocity for each one of the joints, which represents an action of control from the top model. A PID controller is to be tuned for each one of the motors since although they are supposed to be equal, in reality, they have some different parameters and overall condition after being used. Also, because the amplifying circuits for the optical sensors used in the feedback signal were hand manufactured, this will influence the PID tuning for each one of the three loops. Another relevant difference is that the two first motors receive power supply form the first *DFRobot^{TM} 2x2A DC Motor Shield*, and only the third motor is connected to the second shield, as explained in Chapter 5.3.

This shouldn't be very relevant, but actually, this means that the third motor receives a little more power than the other two, since it absorbs the whole power supply of one single shield. Also, the current state of the cables, pins, electronic welds and internal circuit boards' channels of the Arduino and the shields, will deteriorate with time, and they were already second-hand when used in the assembly.

Also visible in the proposed model, is a pre-saturation of the signal before being sent to the Arduino board, so as to limit the maximum velocity corresponding to the PWM maximum voltage. Because this internal loop is meant to control velocities, and we only have position sensors, a general discrete derivative is applied so as to obtain a numerical approach to the current velocity of each joint.

Lastly, the block 'encodersToAngles' is the transformation from the optic encoder counter sent by the Arduino to actual joint angles, in radians. For this transformation, the ratio of the cable transmission system used is considered as well as the corresponding angle to one slit passage of the encoder disk.

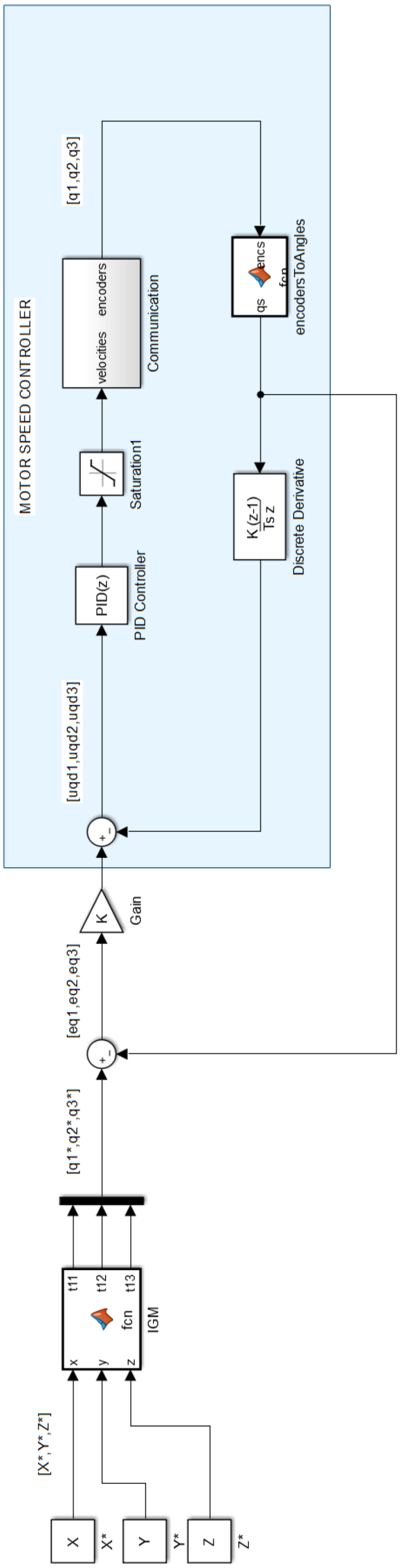


Figure 6.7: Simulink block diagram of the position control model proposed for the manipulator built, with inner motor controllers.

7 | Conclusions

This is the final chapter where the results obtained will be discussed as well as the obstacles faced and solutions found during the length of the project.

7.1 Achievements

When analysing this project, it is possible to see that there was a lot of ambition from the start. The proposed topic was the modelling, kinematic optimization, construction and control of a robot. When there is great ambition, there is also strong will and motivation, and thus one can say that overall, the work was fulfilled and the proposed concept was proved. One may consider it as a prototyping of all these different areas, which resulted in a very practical work. It had a lot of small details and difficulties to be solved that can not be thoroughly expressed in this scientific report, but which consumed more than the double of the time needed to prepare the theoretical part.

Also, because there was the need to do so much research on this many different topics, from kinematic parallel modelling to electronic circuits, passing through C++ and object oriented programming, 3D printing, opAmp theory, serial communication, electronic welding and control, a lot of the work done was not entirely used in the final prototype. This was mainly because, in this kind of practical projects, one problem never comes alone, and only one of the attempts to solve it will succeed, after months of trying dead-end solutions.

Revisiting Chapter 1.5, it is possible to verify how much of the proposed work was achieved. The data was thoroughly selected, filtered and analysed with the help of Matlab computing tools. The robot was accurately modelled for different kinds of mechanical analysis, from statics to dynamics, and these models were all also implemented in Matlab. A feasibility control solution was tested with the original Falcon and its results were presented.

After this, an optimization was conducted in order to enlarge the workspace of the robot, specifically for the desired task. A lot of options were considered at this phase, and many optimizing methods implemented, although not all of them produced good results. With the final result of the optimization process, the design phase began where the robot was extensively 3D modelled and analysed. Also, because of the low resources availability, a lot of creativity and make-do was used with effort to build what was necessary to manufacture, and find the best places to buy what couldn't be built.

Later on, much time was spent trying to understand how the low level, hard-coded SDK of the original micro-controller worked, and if it was possible to modify it to the requirements of the new robot. When this alternative was found to be not possible, the Arduino board was carefully studied and shields were selected to power and control the original motors of the Falcon.

A communication serial system was developed to communicate with the arduino board and to receive the feedback from the optical sensors. Also, these sensors had custom made circuits to allow the Arduino board to properly count the rotations of the encoders' disks, through the digital pins.

Finally, taking into account the feasibility assessment of the initial control study, a control model was developed. This model proposes the implementation of custom tuned PID controllers for each one of the motor's velocity control, and also, a general position control loop that uses the Inverse Geometric Model to control the robot in the joint space and guide it through a defined cartesian trajectory.

7.2 Future Work

In this final section, it is important to look into the future work that this project may have allowed to give continuity. One of the most important and relevant future tasks, will be to continue the implementation of the control model.

It is necessary to further investigate the possibilities revolving the control solution. Since this robot is to be used in a real-time, image feeding system from a TRUS and MRI based fusion, it is necessary to guarantee the appropriate frequency of communication, and also, the possibility to follow given trajectories. Many different kinds of controllers can be studied, such as a more complex controller for the trajectory model, possibly with integrative and derivative terms, in order to improve the system's response. Another option would be to define a state space representation and implement optimal control or even, an optimal observer model.

In this same subject, another future work should be the integration with the ultra-sound probe's electronics. Previous works have been done in image analysis and prostate reconstruction from MRI images, but it is necessary to develop a real-time recognition program to input desired positions into the robot's control system.

Also, it is arguable that the kinematic structure could be further developed in the sense of having more degrees of freedom, and maybe a larger workspace. However, this work was done such that a stiff and robust robot was obtained for minimizing the human-robot interaction issues in medical interventions. So, the actual kinematic structure shouldn't be easily discarded, because of the said benefits.

As referred in the previous paragraph, the advantages of this kinematic structure make it very versatile and appropriate to many different kind of industries. Parallel robots such as this one have been used in the fastest pick and place tasks in the world, but also in fine, precision surgeries where a very rigid and robust manipulator is needed. So, this work ends with the proposition to adapt in future projects, this same prototype for other kinds of surgical interventions, such as brain surgery, for its required precision movements.

Bibliography

[1] P. C. UK. Know your prostate; How prostate cancer is diagnosed. England, 2016. Available at http://prostatecanceruk.org/media/2494030/how-prostate-cancer-is-diagnosed_ifm.pdf (last accessed on 03/12/2016).

[2] J. Regan, V. Cinnamon, and A. Russo. *Seeley's Essentials Of Anatomy & Physiology*. McGraw-Hill Education, 9 edition, 2016. ISBN:978-0-07-809732-4.

[3] J. Ferlay, I. Soerjomataram, M. Ervik, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. Parkin, D. Forman, and F. Bray. Globocan 2012 v1.1, cancer incidence and mortality worldwide: Iarc cancerbase no. 11. International Agency for Research on Cancer, Lyon, France, 2014. Available at http://globocan.iarc.fr (last accessed on 21/10/2016).

[4] W. H. Organization. Estimated prostate cancer mortality worldwide in 2012, per 100 000. GLOBOCA, 2012. Available at http://globocan.iarc.fr/Pages/fact-sheets-cancer (last accessed on 03/12/2016).

[5] D. A. Mandal. Psa false positive and false negative results. 2014. Available from: http://www.news-medical.net/health/ (last accessed on 27/10/2016.).

[6] A. Sciarra, J. Barentsz, A. Bjartell, J. Eastham, H. Hricak, V. Panebianco, and J. A. Witjes. Advances in Magnetic Resonance Imaging : How They Are Changing the Management of Prostate Cancer. *European Urology*, 59(6):962–977, 2011. ISSN 0302-2838. doi: 10.1016/j.eururo.2011.02.034.

[7] D. Tavares. *MRI / TRUS data fusion - Prostate cancer diagnostic and focal treatment*. PhD thesis, Instituto Superior Técnico, 2012.

[8] J. D. Ramalhinho. *Implementation of a 3D TRUS acquisition system for robotized focal biopsies of the prostate - An approach using 3D integrated US transducers*. PhD thesis, Instituto Superior Técnico, 2014.

[9] D. Beyersdorff, A. Winkel, B. Hamm, S. Lenk, S. A. Loening, and M. Taupitz. MR Imaging-guided Prostate Biopsy with a Closed MR Unit at 1.5 T : Initial Results. *Radiology*, 234(2):576–81, 2005. doi: 10.1148/radiol.2342031887.

[10] J. Bax, D. Cool, L. Gardi, K. Knight, D. Smith, J. Montreuil, S. Sherebrin, C. Romagnoli, and A. Fenster. Mechanically assisted 3D ultrasound guided prostate biopsy system. *Medical Physics*, 35(12), 2008. doi: 10.1118/1.3002415.

[11] A. Fenster, J. Bax, H. Neshat, N. Kakani, and C. Romagnoli. 3D Ultrasound Imaging in Image-Guided Intervention. In G. Gunarathne, editor, *Advancements and Breakthroughs in Ultrasound Imaging*, chapter 1. InTech, 2013. doi: 10.5772/55230.

[12] K. Chinzei, N. Hata, F. Jolesz, and R. Kikinis. MR Compatible Surgical Assist Robot: System Integration and Preliminary Feasibility Study. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 3:921–930, 2000. doi: 10.1007/978-3-540-40899-4_95.

[13] A. Krieger, R. C. Susil, C. Ménard, J. A. Coleman, G. Fichtinger, E. Atalar, and L. L. Whitcomb. Design of a Novel MRI Compatible Manipulator for Image Guided Prostate Interventions. *IEEE Transactions on Biomedical Engineering*, 52(2):306–313, 2005. doi: 10.1109/TBME.2004.840497.

[14] P. C. Mozer, A. W. Partin, and D. Stoianovici. Robotic Image-Guided Needle Interventions of the Prostate. *Reviews in urology*, 11(1):7–15, 2009. ISSN 1523-6161.

[15] D. Yakar, M. G. Schouten, D. G. H. Bosboom, J. O. Barentsz, T. W. J. Scheenen, and J. J. Fütterer. Feasibility of a Pneumatically Actuated MR-compatible Robot for Transrectal Prostate Biopsy Guidance. *Radiology*, 260(1):241–247, 2011. doi: 10.1148/radiol.11101106. PMID: 21406625.

[16] H. Su, D. C. Cardona, W. Shang, A. Camilo, G. A. Cole, D. C. Rucker, R. J. Webster, and G. S. Fischer. A mri-guided concentric tube continuum robot with piezoelectric actuation: A feasibility study. pages 1939–1945, May 2012. ISSN 1050-4729. doi: 10.1109/ICRA.2012.6224550.

[17] H. Ho, P. Mohan, E. Lim, D. Li, J. Yuen, W. Ng, W. Lau, and C. Cheng. Robotic ultrasound-guided prostate intervention device: system description and results from phantom studies. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 5:51–58, 2009. doi: 10.1002/rcs.232.

[18] C. Poquet, P. Mozer, M. A. Vitrani, and G. Morel. An Endorectal Ultrasound Probe Comanipulator With Hybrid Actuation Combining Brakes and Motors. *IEEE/ASME Transactions on Mechatronics*, 20(1):186–196, 2015. doi: 10.1109/TMECH.2014.2314859.

[19] N. D. Inc. Passive Polaris Spectra User Guide, 2012.

[20] M. Humbert, N. Grey, J. Muller, and C. Esling. Determination of a mean orientation from a cloud of orientations. application to electron back-scattering pattern measurements. *Journal of Applied Crystallography*, 29:662–666, 1996.

[21] N. T. Inc. Novint Falcon User Manual, 2007.

[22] R. Stamper. A three degree of freedom parallel manipulator with only translational degrees of freedom. *Drumstage.Lib.Umd.Edu*, page 211, 1997. doi: 10.1.1.24.7361.

[23] A. V. Shah, S. Teuscher, E. W. Mcclain, and J. J. Abbott. *How to Build an Inexpensive 5-DOF Haptic Device Using Two Novint Falcons*, pages 136–143. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-14064-8. doi: 10.1007/978-3-642-14064-8_21.

[24] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer Publishing Company, Incorporated, 1 edition, 2008. ISBN 978-1-84628-641-4. doi: 10.1007/978-1-84628-642-1.

[25] T. Yoshikawa. *Foundations of Robotics: Analysis and Control*. MIT Press, Cambridge, MA, USA, 1990. ISBN 0-262-24028-9.

[26] D. J. Block, M. B. Michelotti, and R. S. Sreenivas. Application of the novint falcon haptic device as an actuator in real-time control. *Paladyn, Journal of Behavioral Robotics*, 4(3):182–193, 2013. doi: 10.2478/pjbr-2013-0017.

[27] S. Martin and N. Hillier. Characterisation of the novint falcon haptic device for application as a robot manipulator. In *Australasian Conference on Robotics and Automation (ACRA*, Syndney, Australia, December 2009.

[28] R. Byrd, J. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000. ISSN 0025-5610. doi: 10.1007/s101070000189.

[29] K. Machulis. Everything i know about the novint falcon as of march 2008, March 2008. Available at `https://kyle.machul.is/2008/03/25/everything-i-know-about-the-novint-falcon-as-of-march-2008/` (last accessed on 10/02/2017).

[30] A. Arduino. Arduino uno technical specs, 2017. Available at `https://www.arduino.cc/en/Main/arduinoBoardUno#techspecs` (last accessed on 19/02/2017).

[31] D. Robot. Arduino motor shield (l298n) (sku:dri0009), 2017. Available at `https://www.dfrobot.com/wiki/index.php/Arduino_Motor_Shield_(L298N)_(SKU:DRI0009)` (last accessed on 19/02/2017).

[32] A. Arduino. Arduino constants, 2017. Available at `https://www.arduino.cc/en/Reference/Constants` (last accessed on 12/02/2017).

[33] M. Schwager. Enableinterrupt library, 2015. Available at `https://github.com/GreyGnome/EnableInterrupt/blob/master/` (last accessed on 25/02/2017).

[34] M. D. Pascale and D. Prattichizzo. The haptik library. *IEEE Robotics Automation Magazine*, 14(4): 64–75, Dec 2007. ISSN 1070-9932. doi: 10.1109/M-RA.2007.905747.

A | Geometric Jacobian Differentiation

To compute the derivative of the kinematic jacobian, Eq. 3.24 is differentiated with respect to time.

$$\dot{\mathbf{J}} = \frac{d}{dt}(\mathbf{A}_r)^{-1}\mathbf{B} + (\mathbf{A}_r)^{-1}\frac{d}{dt}\mathbf{B} \tag{A.1}$$

Knowing that the inverse of A_r is defined as follows:

$$(\mathbf{A}_r)^{-1} = \frac{1}{\det(A_r)} adj(A_r) \tag{A.2}$$

Where $adj(A_r)$ is the adjoint matrix of A_r and $\det(A_r)$ its determinant. These are defined in Equations A.4 and A.3, respectively.

$$\det(\mathbf{A_r}) = A_{11}A_{22}A_{33} - A_{11}A_{23}A_{32} - A_{12}A_{21}A_{33} + A_{12}A_{23}A_{31} + A_{13}A_{21}A_{32} - A_{13}A_{22}A_{31} \tag{A.3}$$

The adjoint matrix can be computed as the transpose of the cofactor matrix. The calculations are a bit extensive and only the final result is presented.

$$adj(\mathbf{A_r}) = \begin{bmatrix} A_{22}A_{33} - A_{23}A_{32} & A_{13}A_{32} - A_{12}A_{33} & A_{12}A_{23} - A_{13}A_{22} \\ A_{23}A_{31} - A_{21}A_{33} & A_{11}A_{33} - A_{13}A_{31} & A_{13}A_{21} - A_{11}A_{23} \\ A_{21}A_{32} - A_{22}A_{31} & A_{12}A_{31} - A_{11}A_{32} & A_{11}A_{22} - A_{12}A_{21} \end{bmatrix} \tag{A.4}$$

Afterwards, it is necessary to find the derivative of $(\mathbf{A}_r)^{-1}$ defined as:

$$\frac{d}{dt}(\mathbf{A}_r)^{-1} = \frac{1}{\det^2(\mathbf{A_r})}\left(\frac{d}{dt}\Big(adj(\mathbf{A_r})\Big)\det(\mathbf{A_r}) - adj(\mathbf{A_r})\frac{d}{dt}\Big(\det(\mathbf{A_r})\Big)\right) \tag{A.5}$$

Where the derivative of the determinant of Equation A.3 can be computed such that:

$$\begin{aligned}\frac{d}{dt}\det(\mathbf{A_r}) = &\Big(\dot{A}_{11}A_{22}A_{33} + A_{11}\;{}^{22}\Omega_{33}\Big) - \Big(\dot{A}_{11}A_{23}A_{32} + A_{11}\;{}^{23}\Omega_{32}\Big) - \Big(\dot{A}_{12}A_{21}A_{33} + A_{12}\;{}^{21}\Omega_{33}\Big) \\ &+ \Big(\dot{A}_{12}A_{23}A_{31} + A_{12}\;{}^{23}\Omega_{31}\Big) + \Big(\dot{A}_{13}A_{21}A_{32} + A_{13}\;{}^{21}\Omega_{32}\Big) - \Big(\dot{A}_{13}A_{22}A_{31} + A_{13}\;{}^{22}\Omega_{31}\Big)\end{aligned} \tag{A.6}$$

And the derivative of the adjoin matrix is defined as:

$$\frac{d}{dt}adj(\mathbf{A_r}) = \begin{bmatrix} {}^{22}\Omega_{33} -{}^{23}\Omega_{32} & {}^{13}\Omega_{32} -{}^{12}\Omega_{33} & {}^{12}\Omega_{23} -{}^{13}\Omega_{22} \\ {}^{23}\Omega_{31} -{}^{21}\Omega_{33} & {}^{11}\Omega_{33} -{}^{13}\Omega_{31} & {}^{13}\Omega_{21} -{}^{11}\Omega_{23} \\ {}^{21}\Omega_{32} -{}^{22}\Omega_{31} & {}^{12}\Omega_{31} -{}^{11}\Omega_{32} & {}^{11}\Omega_{22} -{}^{12}\Omega_{21} \end{bmatrix} \tag{A.7}$$

Where,

$${}^{ij}\Omega_{kl} = \dot{A}_{ij}A_{kl} + A_{ij}\dot{A}_{kl} \tag{A.8}$$

And,

$$\dot{A}_{i1} = \Bigg(- \sin(\theta_{2i})\sin(\theta_{3i})\dot{\theta}_{2i} + \cos(\theta_{2i})\cos(\theta_{3i})\dot{\theta}_{3i} \Bigg) \cos(\phi_i) + \sin(\theta_{3i})\dot{\theta}_{3i}\sin(\phi_i) \tag{A.9}$$

$$\dot{A}_{i2} = \Bigg(- \sin(\theta_{2i})\sin(\theta_{3i})\dot{\theta}_{2i} + \cos(\theta_{2i})\cos(\theta_{3i})\dot{\theta}_{3i} \Bigg) \sin(\phi_i) - \sin(\theta_{3i})\dot{\theta}_{3i}\cos(\phi_i) \tag{A.10}$$

$$\dot{A}_{i3} = \cos(\theta_{2i})\sin(\theta_{3i})\dot{\theta}_{2i} + \sin(\theta_{2i})\cos(\theta_{3i})\dot{\theta}_{3i} \tag{A.11}$$

$$\dot{B}_i = a\cos(\theta_{2i} - \theta_{1i})\sin(\theta_{3i})(\dot{\theta}_{2i} - \dot{\theta}_{1i}) + a\sin(\theta_{2i} - \theta_{1i})\cos(\theta_{3i})\dot{\theta}_{3i} \tag{A.12}$$

Substituting Equations A.2 and A.5 in Equation A.1, one obtains:

$$\dot{\mathbf{J}} = \frac{\mathbf{B}}{\det^2(\mathbf{A_r})}\left(\frac{d}{dt}adj(\mathbf{A_r})\det(\mathbf{A_r}) - adj(\mathbf{A_r})\frac{d}{dt}\det(\mathbf{A_r})\right) + \frac{\dot{\mathbf{B}}}{\det(A_r)}adj(A_r) \tag{A.13}$$

Where,

$$\dot{\mathbf{B}} = \begin{bmatrix} \dot{B}_1 & 0 & 0 \\ 0 & \dot{B}_2 & 0 \\ 0 & 0 & \dot{B}_3 \end{bmatrix} \tag{A.14}$$